İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

BLIND AUDIO SOURCE SEPARATION USING NONNEGATIVE TENSOR FACTORIZATION TECHNIQUES

M.Sc. Thesis by

M. Altug KEYDER, B.Sc.

Department : **Electronics and Telecommunications Engineering**

Programme: **Telecommunications Engineering**

JANUARY 2008

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

BLIND AUDIO SOURCE SEPARATION USING NONNEGATIVE TENSOR FACTORIZATION TECHNIQUES

M.Sc. Thesis by

M. Altug KEYDER, B.Sc.

504051317

Date of submission : 24 December 2007

Date of defence examination : 29 January 2008

Supervisor (Chairman): Prof.Dr. Bilge GÜNSEL

Members of the Examining Committee Prof.Dr. Muhittin GÖKMEN

Assoc.Prof.Dr. Işın ERER

JANUARY 2008

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

NEGATİF OLMAYAN TENSÖR FAKTÖRİZASYONU KULLANILARAK GÖZÜ KAPALI SES KAYNAK AYRIŞTIRMA

YÜKSEK LİSANS TEZİ

M. Altuğ KEYDER, Müh.

504051317

Tezin Enstitüye Verildiği Tarih : 24 Aralık 2007

Tezin Savunulduğu Tarih : 29 Ocak 2008

Tez Danışmanı : Prof.Dr. Bilge GÜNSEL

Diğer Jüri Üyeleri : Prof.Dr. Muhittin GÖKMEN

Assoc.Prof.Dr. Işın ERER

JANUARY 2008

PREFACE

Here, I would like to thank to everyone who helped, encouraged and supported me to start and finish this work successfully. Especially to my advisor Prof. Dr. Bilge Gunsel, my family and friends.

January 2008 M. Altug KEYDER

CONTENT

ABBREVIATIONS

ALS	: Alternating Least Squares
BSS	: Blind Source Separation
FFT	: Fast Fourier Transform
FPALS	: Fixed Point Alternating Least Squares
ICA	: Independent Component Analysis
ITU	: International Telecommunication Union
MOV	: Model Output Variables
NMF	: Non-negative Matrix Factorization
NTF	: Non-negative Tensor Factorization
PARAFAC	: Parallel Factor Analysis
PCA	: Principal Component Analysis
RALS	: Regularized Alternating Least Squares
SIR	: Signal to Interference Ratio
SVD	: Singular Value Decompostion

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$\underline{X}$	: Tensor X or n-way array
X	: Matrix X or 2-way array
$\mathbf{X}$	: Matrix created by unfolding the tensor $\underline{X}$
$\mathbf{x}$	: Vector $\mathbf{x}$ or 1-way array
x_{ijk}	: ijk-th element of tensor $\underline{X}$
x_{ij}	: ij-th element of the matrix X
x_i	: i-th element of the vector $\mathbf{x}$
$\Re_+$	: Non-negative orthant space

NEGATİF OLMAYAN TENSÖR FAKTÖRİZASYONU KULLANILARAK GÖZÜ KAPALI SES KAYNAK AYRIŞTIRMA

ÖZET

Bu çalışmada, kokteyl partisi problemi olarak da bilinen, gözü kapalı ses işareti ayrıştırma probleminin negatif olmayan tensor faktörizasyonu yöntemi kullanılarak nasıl çözüldüğü araştırılmıştır. Gözü kapalı kaynak ayrıştırma (BSS) problemi genel olarak, hakkında önceden bilgi sahibi olmadığımız kaynak işaretlerininin lineer karışımlarından oluşan gözlemlerden, kaynak işaretlerin kestirilmesi(ayrıştırılması) işlemidir. Burada 'gözü kapalı' ibaresinin kullanılmasının sebebi, kaynak işaretleri hakkında hiçbir ön bilgiye sahip olunmamasıdır, ancak kaynak işareti hakkında zayıf varsayımlar yapılabilinir.

Bugüne kadar gözü kapalı kaynak ayrıştırma probleminin çözümü için birçok yöntem önerilmiştir, bunların başında Bağımsız Bileşen Analizi (Independent Component Analysis, ICA), Tekil Değer Ayrıştırması (Singular Value Decompostion, SVD) gibi yöntemler gelmektedir. Bunların yanı sıra yeni bir yaklaşım olan Negatif Olmayan Tensör/Matris Ayrıştırması (NTF/NMF) yöntemi ise giderek araştırmacıların dikkatini çeken bir yöntem olmaya başlamıştır. Hem NMF hem de NTF temelde kaynakların ve gözlemlerin negatif olmayan değerlerle ifade edilebileceği versayımına dayanmaktadır. Aralarındaki farklılık ise, verinin(kaynak ve veya gözlem) ifade edildiği uzayın boyuttur. Buna göre; veri iki boyutla ifade edildiği taktrirde matris ayrıştırması, ikiden fazla boyutla ifade edildiği taktirde ise tensör(çok boyutlu matris) ayrıştırması yapılmaktadır.

Bu çalışmada üç farklı NTF yönteminin ses işaretleri üzerindeki ayrıştırma başarımları incelenmiştir. Bu üç yöntemde de ötelemeli olarak uygulanan ve 'eğimli azalama (gradient descent)' gibi bilindik optimizasyon yöntemleri kullanılarak türetilmiş güncelleme kuralları kullanılır. Aralarındaki temel farklılık ise, optimizasyon için seçilmiş olan maliyet fonksiyonları asındaki değişikliklerden kaynaklanmaktadır. Bu yöntemler; değişimli en küçük kareler algoritması (Alternating Least Squares, ALS), ve sırasıyla alfa ve beta olarak bilinen maliyet fonksiyonları kıllanılarak oluşturulmuş alfa ve beta algoritmalarıdır. Her bir algoritmanın ayrıştırma başarımları, farklı lineer karışımlar, gürültülü karışımlar ve farklı ilk koşullar gibi bir çok test koşulu altında denenmiştir. Genel olarak varılan noktada gözlemlenmiştir ki, NTF yöntemleri kullanılarak gözü kapalı kaynak ayrıştırma probleminin çözümünde başarılı sonuçlar elde edilmiştir. Bu üç algoritma arasında yapılan karşılaştırmalar göstermiştir ki, ALS algoritması bütün koşullar altında daha yüksek başarım sergilemiştir. Belirtilmesi gereken bir diğer durum da, beta algoritmasının uygun parameteler altında koşturulduğu taktirde ALS algoritmasına yakın başarım gösterebildiğidir. Ancak işlemsel karmaşıklık açısından bakıldığı taktirde de, ALS algoritmasının diğer iki algoritmadan üstün olduğu görülmüştür.

BLIND AUDIO SOURCE SEPARATION USING NONNEGATIVE TENSOR FACTORIZATION TECHNIQUES

SUMMARY

In this work, the success of Nonnegative Tensor Factorization on the solution of Blind audio source separation (ABSS) problem which is also known as 'cocktail party problem' is studied. BSS in general, is the process of recovering a set of signals, which are called source signals, from a set of mixture of those signals which are called the observations. The term 'blind' refers to that neither the charcteristics of the source nor the mixing process is known. i.e.: no a priori information. Not only in audio signals but also in many fields of signal processing, BSS takes place.

There are several methods, proposed to solve the BSS problem such as Independent Component Analysis (ICA), Singular Value Decomposition(SVD). One of the most recently proposed approach is called Nonnegative Tensor/matrix factorization (NTF/NMF). Both NMF and NTF depends on the assumption that both the source signals that are supposed to be estimated and the mixture signals are represented by nonnegative numbers. The diffence between NMF and NTF is the dimension which is used to represent data. Meaning that, for two dimensional representation of mixture signals NMF can be used, on the other hand for more than two dimensions the tensor factorization concept must be introduced.

In this very research the separation performance of the three important NTF methods are studied. All three methods depends on iterative update rules which are derived by using common optimization methods such as gradient descent. The difference among these methods is the cost functions that are used to derive the update rules. These three algotihms are; the alternating least squares (ALS) algorithm, alpha and beta algorithms which are obtained by employing gradient descent on the cost functions called α-divergence and β-divergence, respectively.

The separation performance of the algorithms are tested under several conditions such as noisy mixtures, different initializations of the algorithms, different mixing conditions. It is observed that, in general the NTF methods yield quite promising results in BSS problem. More specifically the ALS and its regularized form perform better separation than alpha and beta algorithms. It should be noted that, the performance of the beta algorithm can be improved if the parameters of the algorithm are selected carefully. However from the computational complexity point of view, the ALS algorithm is still superior.

1. INTRODUCTION

In various kinds of fields in signal processing such as speech recognition, biomedical signal processing, video processing, the fundamental problem is how to represent large data sets in an efficient way. Data representation based on decomposition is one of the simpliest approaches to overcome this problem. In these methods, the data set which involves various signals can be decomposed into two factors which generally have lower dimensions but still conserving enough information to represent the content of data itself. Blind source separation (BSS) is the process of recovering a set of signals, which are called source signals, from a set of mixture of those signals, which are called the observations. The term 'blind' refers to that neither the charcteristics of the source nor the mixing process is known. i.e.: no a priori information. Typically only some weak assumptions are made about the sources and mixing process. The BSS problem can be defined as a decomposition problem where the decomposed factors are unknow.

There are different successful methods that are proposed recently to solve BSS problems. [1,2] Principle Component Analysis (PCA) which is also known as Karhunen-Loeve transform is a well known method for dimensionality reduction that transform data to another coordinate system by maximazing the variance of the basis[1] [3]. Another one is Independent Component Analysis (ICA) which assumes that the source signals are mutually independent and non-Gaussian [4]. Unlike PCA, ICA uses higher order statistics to separate data, i.e., by maximizing the statistical independence, the independent components (aka sources) are found. Singular Value Decomposition (SVD) which is another important matrix factorization method, based on the spectral theorem that says normal matrices can be unitarily be diagonalized using a basis of eigenvectors [5]. Nonnegative Matrix/Tensor Factorization (NMF/NTF) which is a general name for a set of methods that are used

[1] In PCA basis are not fixed unlike other orthogonal linear transforms, rather depend on the data set.

to factorize a data matrix X into two matrices A and S subjected to constraint that both A and S are nonnegative. The nonnegative matrix factorization was first studied by Finnish group of researchers with the name of positive matrix factorization [6,7]. The name of this method became nonnegative matrix factorization after the studies of Lee and Seung [8,9] who proposed a simple and effective algorithms for this factorization. The basic Nonnegative Tensor Factorization (NTF) method which is actually the constrained version of well known method called Parallel Factor Analysis [10-19] (PARAFAC), can be defined as an extension of NMF from 2-way arrays to n-way arrays (aka tensors). Distinction between various NMF algorithms comes generally from the used cost functions and applied regularization approaches.

Even though these methods make weak assumptions to overcome the pitfalls of the BSS problem, they have high performace on reconstructing the original signals from observations.

The aim of this work is to investigate the success of nonnegative tensor factorization method [10-19] in blind audio source separation. The blind audio source separation, recovering audio source signals from their linear mixtures, is also refered to as 'cocktail party problem'. In this problem several people, talking in the same room, cause their voice to mix each other. The human hearing system has to separate those mixtures in order to follow particular speaker in the room. Even though this problem is handled easily by human brain, this is not the case in automatic source separation by applying a digital signal processing method. From the digital signal processing point of view the speakers can be considered as different simultaneously active audio sources and transmission through the room as the mixing process and the recordings made by microphones, placed in different spatial locations in the room, as the observations (mixtures).

Various NTF approaches which involve different matricizing[2] and optimization techniques [14-16,20] are studied and compared with each other to solve the blind source separation problem. The NTF and nonnegatively constrained parallel factor analysis are two important methods which are recently proposed as sparse and

[2] Unfolding tensor into matrices to create analogy with NMF.

efficient representations of signals [14-18] The advantage of NTF over two dimensional matrix factorizations such as NMF is that NTF takes into account spacial and temporal correlations between variables more accurately. This is why we have preferred using NTF, in this work, rather than other methods.

Based on the propsed methods [14-16,20] for NTF, a software is desinged and several tests are performed. The results are evaluated and a comparison of these methods are made from the blind audio source separation point of view. It has been observed that performace of the NTF based algorithms are quite promising in blind audio source separation and even more successful than NMF. Among the implemented three different algorithms, it is seen that the performance of Alternating Least Squares (ALS) algorithm, with and without the regularization, is superior than the beta-divergence and alpha-divergence algorithms [14,18]. It should be noted that the performance of the beta-divergence algorithm can be competitive with the ALS algorithm if the parameters can be chosen carefully.

There are several test cases that can be used to evaluate the performance of a blind source separation algorithm. Test cases that are taken into account in this study involve; evaluation of robustness to initial conditions, different mixing matrices, and additive noise. parameter selection, presence of additive noise, using different types of source signals (i.e.: speech, music, single tone sinusoids). In each test, maximum of two sources signals are used and mixtures are obtained using randomly generated mixing matrices. All the tests are performed via a software which is designed using Java programming language. This software is capable of performing preprocessing on source signals, mixing source signals, running several NTF algorithms on mixed signals and reporting results. As a measure of algorithm performance, two criterion are chosen, Amari index which is commonly used in the literature [25] and the Objective Perceptual Audio Quality measurement criterions, defined in the recommendation report, ITU-R BS.1387 of International Telecommunication Union (ITU). The traditional approach for performance evaluation consider only the signal to interference ration (SIR) and Amari index [23]. However in this work, it is shown that recently standardized perceptual audio quality measurement criteria is also important and should be taken into account for performance evaluation of the separation algorithms.

The thesis consists of four headlines, In Section 2 , following the introduction, the nonnegative tensor factorization models are stated in detail. These are called NTF-1 and NTF-2 models [15,16]. The problem formulation of blind source separation are built on these models. Also the optimization algorithms are stated upon these models [15,16]. The three algorithms, used in this work are also explained in detail in this section of the thesis. These algorithms are called Alternating Least Squares (ALS), Beta-divergence and Alpha-divergence algorithms [15,16]. Section 3 presents explanation of the software which is designed to realize blind audio source separation based on nonnegative tensor factorization. This part involves the explanation of graphical user interface of the software and the properties of the software. Following this part, the test cases are defined and the corresponding test results are given. Comparison between performances of the desıgned algorithms has aldo been made in Section 3. On the final section, the conclusion of the work is stated by commenting on the test results given in the previous section. Also the comments on the future work are given. At the very back of this thesis the References and the Appendix parts take place. It is also important to note that the waveforms of the mixtures (observations) which are used througout the tests are given in the Appendix A.

2. AUDIO SOURCE SEPARATION BY NON-NEGATIVE TENSOR FACTORIZATION

The basic Non-negative Tensor Factorization (NTF) model can be described as an extansion of the well known Parallel Factor Analysis (PARAFAC) method introduced in [10,16]. In both methods, the aim is to decompose multi-way arrays except that NTF imposes a nonnegativity constraint on the decomposed components.

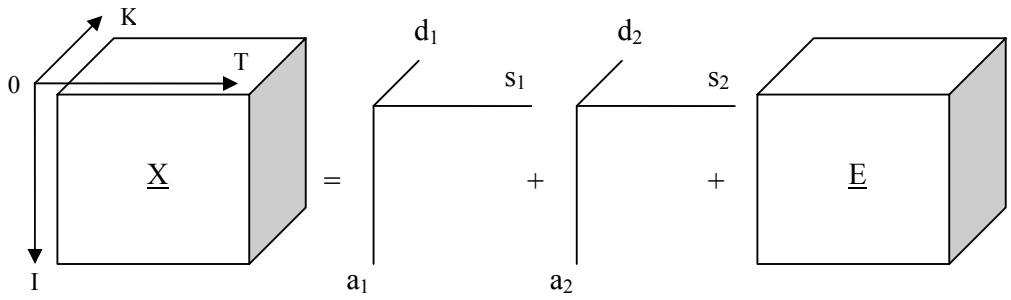

Figure 2.1 : Graphical representation of a two-component decomposition model.

Mathematical representation of a two-component decompostion model for three-way arrays is given by Eq.(2.1)

$$x_{itk} = \sum_{r=1}^{R=2} a_{ir} s_{tr} d_{kr} + e_{itk}\,,\ \text{i} = 1,...,\text{I},\ \text{t} = 1,...,\text{T},\ \text{k} = 1,...,\text{K} \qquad \textbf{(2.1)}$$

where R, the upper limit of summation represents rank of the decomposition. In Eq.(2.1), x_{itk} refers to the *itk*-th element of the tensor, $\underline{X}$, a_{ir}, s_{tr}, d_{kr} and e_{itk} are elements of the corresponding tensors and vectors, respectively. Corresponding graphical representation is shown in Fig.2.1.

Eq.(2.2) defines the corresponding tensor-vector notation,

$$\underline{X} = \sum_{r}^{2} a_r \otimes s_r \otimes d_r + \underline{E}, \qquad \textbf{(2.2)}$$

where a_r, s_r, and d_r are the *r*-th columns of matrices A, S, and D, respectively. From blind source separation point of view, $\underline{X}$ is the tensor which represents observed data, the matrix A is the nonnegative mixing matrix representing common factors (basis), S is the nonnegative matrix of sources, D is the nonnegative diagonal scaling matrix and E is the tensor which represents decomposition (estimation) error or additive environmental noise which is generally inherent in all acoustic mixing conditions.

One of the advantages of the described 3-D factorization method over 2-D factorization methods such as NMF, is that it conserves the spatial and temporal structure of the observed data. The other important advantage of the basic model, definded above, is the uniqueness[3] of the solution.[14,16,17].

Assuming that the sources are mixed linearly, the blind source separation problem can be fit into mathematical model by using the following simple expression,

$$X = AS + E. \qquad \textbf{(2.3)}$$

Generally speaking, X is the matrix[4] representing observations, A is the mixing matrix, S is the sources and E is the additive noise. Both A and S are unknown and the goal is to find either A or S, since once one of the two is found the other can be solved easily. In the literature, this problem is named as blind audio source separation and several algorithms are proposed to perform the source decompostion. Some of the important approaches are explained in detail in the following sections.

Throughout this work, factorization of tensors is performed by first unfolding the tensors into matrices and then decomposing these matricized tensors into two other matrices rather than the basic model, mentioned above. The NTF-1 and NTF-2 are two factorization models which can be used to unfold (matricize) tensors into

[3] the decomposition converges to same point even if the input matrix(observations), is rotated.

[4] Here, the scope of term 'matrix' includes not only 2-D arrays but also n-D arrays, hence tensors.

matices[16]. It should be noted that once the tensors are represented by matrices, the algorithms proposed for NMF can also be used in NTF.

2.1 NTF-1 Factorization Model

The 3-D NTF1 model, illustrated in Figure 2.2, attempts to estimate only one common factor in the form of basis matrix A. A given tensor $\underline{X} \in \Re^{I\times T\times K}$ is decomposed to a set of matrices A, D and $\{S_1, S_2,..., S_K\}$with nonnegative entries, where K is the number of frontal slices (number of observations) of the tensor $\underline{X}$, I and T are the dimensions of the each observed data. The mathematical expression which corresponds to this model can be writtens as,

$$X_k = AD_kS_k + E_k, (k = 1, 2,...,K) \quad \textbf{(2.4)}$$

Objective of blind source separation is to estimate the set of matrices A, D and $\{S_1, S_2,..., S_K\}$ subject to some non-negativity constraints and other possible natural constraints such as sparseness and/or smoothness.

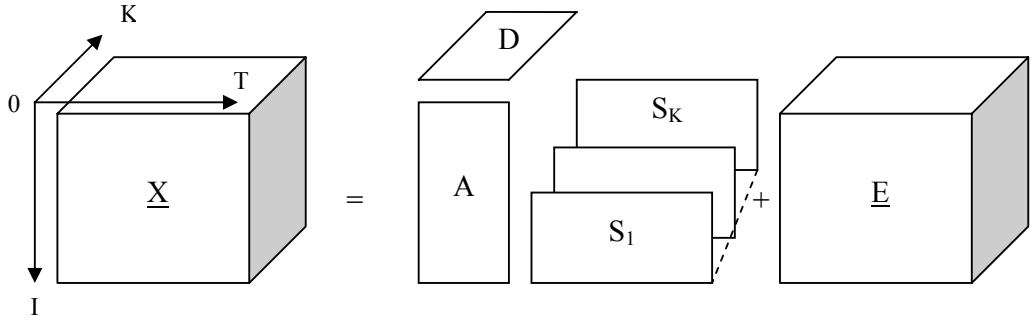

Figure 2.2 : NTF-1 Model

The non-negative diagonal matrices $D_k \in \Re_+^{R\times R}$ are scaling matrices therefore they can usually be merged into the matrices $S_k \in \Re_+^{R\times T}$ by introducing row-normalized matrices $S_k = D_k S_k \in \Re_+^{R\times T}$. Hence, usually the nonnegative matrix A and the set of scaled matrices $\{S_1, S_2,..., S_K\}$need only to be estimated.

The NTF1 model can be converted to row-wise unfolding [16] of a tensor $\underline{X} \in \Re^{I\times T\times K}$ as it is illustrated on Figure 2.3, which could be generally described by a simple matrix equation,

$$\mathbf{X} = [X_1, X_2, ..., X_K] = A\mathbf{S} + \mathbf{E} \qquad \textbf{(2.5)}$$

where $\mathbf{X} = [X_1, X_2, ..., X_K] \in \Re_+^{I\times KT}$ is a row-wise (horizontal) unfolded matrix of all frontal slices $X_k = X_{:,:,k} \in \Re^{I\times T}$, $\mathbf{S} = [S_1, S_2, ..., S_K] \in \Re_+^{R\times KT}$ is a row-wise unfolded matrix of the slices $S_k = S_{:,:,k} \in \Re_+^{R\times T}$ representing nonnegative sources, E = [E1, E2, ..., Ek] $\in \Re^{I\times KT}$ is an unfolded matrix of error.

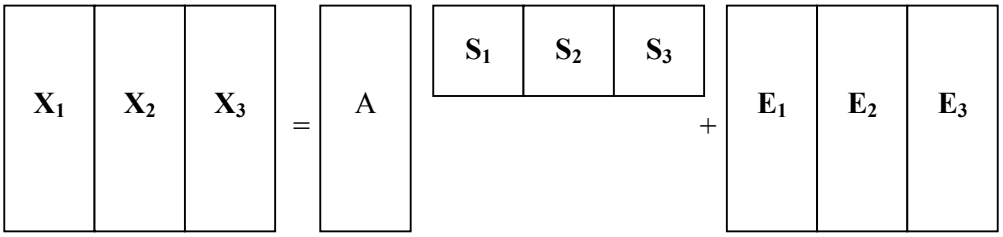

Figure 2.3 : Row-wise unfolded NTF-1 model

2.2 NTF-2 Factorization Model

The 3-D NTF2 model, illustrated in Figure 2.4, can be expressed by a set of matrix equations shown in Eq.(2.6),

$$X_k = A_k D_k S + E_k, (k = 1, 2,...,K) \qquad \textbf{(2.6)}$$

where common factors are represented by the matrix $S \in \Re_+^{R\times T}$, and the basis(mixing) matrices $A_k \in \Re_+^{I\times R}$ are generally different. As it is mentioned for the NTF-1, the diagonal scaling matrices $D_k \in \Re_+^{R\times R}$ can be absorbed by the matrices $A_k \in \Re_+^{I\times R}$ by column normalizing the basis as $A_k = AD_k$. Therefore, in practice only the set of the normalized matrices $\{A_1, A_2, ..., A_K\}$ and S need to be estimated.

Here, it is worth metioning that in both models once this scaling matrix D is merged into other matrices either by column or row normalization, the uniqueness of the model is no longer guaranteed. This situation arise, since scaling and permutation ambiguities are ignored by performing this operation. Simply, X is a martix of observations with each row representing one observation, and by applying simple decomposition, as in Eq.(2.3), its expected that S is the matrix representing estimated sources with each row corresponds to one source. The problem is that, it is not always easy to say which row corresponds to which source, i.e.: the order of estimations may change.

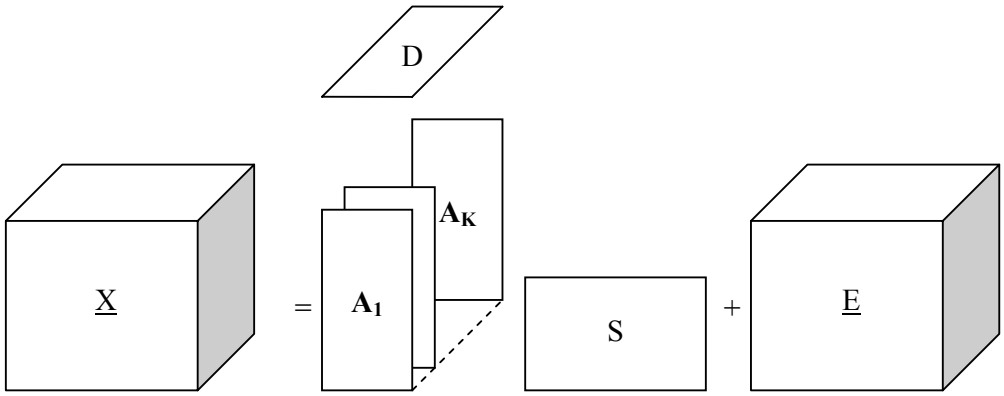

Figure 2.4 : NTF-2 Model

It should be noted that the NTF-2 model can be represented as column wise unfolding as it is illustrated in Figure 2.5.

The unfolding system can be described by a single system of matrix equation,

$$\mathbf{X} = \mathbf{A}S + \mathbf{E}, \qquad \textbf{(2.7)}$$

where $X = [X1; X2 ...; XK] \in \Re_{+}^{KI \times T}$ is a column-wise (vertical) unfolded matrix of the all frontal slices $X_k = X_{:,:,k} \in \Re^{I \times T}$, $\mathbf{A} = [A_1; A_2; ...; A_K] \in \Re_{+}^{KI \times R}$ is a column-wise unfolded matrix of the slices $A_k = A_{:,:,k} \in \Re_{+}^{I \times R}$ representing non-negative basis matrices (the frontal slices), $E = [E1; E2; ...; Ek] \in \Re^{KI \times T}$ is the column-wise unfolded matrix of error or noise.

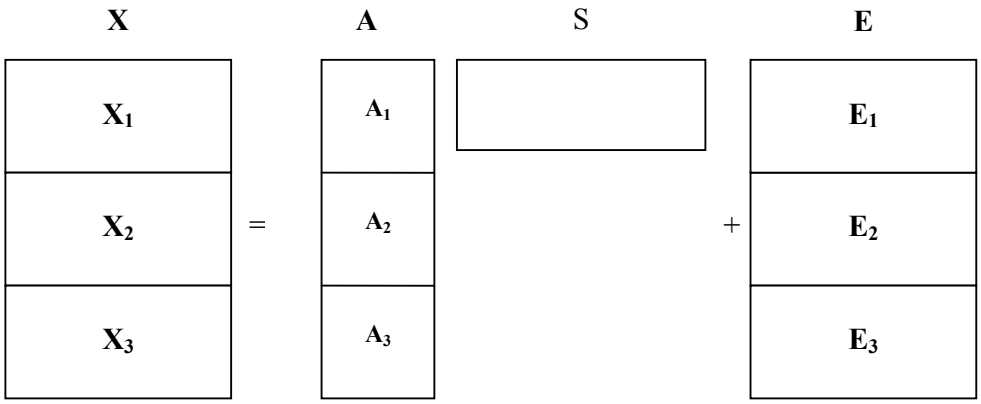

Figure 2.5 : Column-wise unfolded NTF-2 model

The NTF-1 and NTF-2 models are in fact dual of each other, meaning that the same algorithms can be used for both models by taking into account the Eq.(2.8),

$$X_k^T = S^T D_k A_k^T + E_k^T, (k = 1, 2, ..., K). \quad \textbf{(2.8)}$$

3-D NTF models can be transformed to a 2-D non-negative matrix factorization problem by unfolding (matricizing) tensors[16]. However, it should be noted that such a 2-D model in general is not exactly equivalent to a standard NMF model, since we usually need to impose different additional constraints for each slice k. In other words, the unfolded model should be not considered as equal to a standard 2-way NMF of a single 2-D matrix.

2.3 Factorization by Optimization

Generally, the distinction between NTF algoritms arise from the selection of cost functions (also known as divergences in convex optimization) and whether the smoothness and/or sparseness parameters are merged into the cost functions. In this subsection, different optimization methods reported in the literature are explained in detail, starting from the simpliest and famous Lee and Seung method to more complicated methods [8,14,15,16,20]. It should be mantioned that all the following methods are designed for 2-D arrays (matrices), however this does not violate the tensor structure of our problem, since tensors can be represented as matrices after unfolding. Therefore to sustain the simplicity of the notation, in this section of the thesis, the matrices shown in Eq.(2.9) are considered as the unfolded tensors,

$X=AS+E$, s.t. $A\geq 0$, $X\geq 0$ (component-wise), **(2.9)**

where $X \in \Re^{IxT}$, $A \in \Re^{IxR}$ and $S \in \Re^{IxT}$ are the matrices of observations, mixing and unknown sources, respectively.

2.3.1 Optimization by Alternating Multiplicative Updating

Two different optimization cost (loss) functions are considered by Lee and Seung [8,9] : the squared Euclidean distance (squared Frobenius norm) and generalized Kullback-Leibler (KL) divergence, described by Eq.(2.10) and Eq.(2.11), respectively.

$$D_F(X \,||\, AS) = \frac{1}{2}\|X - AS\|_F^2, \quad \textbf{(2.10)}$$

$$D_{KL}(X \,||\, AS) = \sum_{ik}\left(x_{ik} \log \frac{x_{ik}}{[AX]_{ik}} + [AX]_{ik} - x_{ik} \right). \quad \textbf{(2.11)}$$

In [8,9], Lee and Seung proposed two algorithms that perform alternating minimization of the specific cost function using gradient descent. In optimization theory, update rules that perform alternating switching between set of parameters to generate the updates till convergence is referred as "alternating update rules." Unlike the similar algorithms proposed in the literature earlier, Lee and Seung showed that cost function of the NTF optimization problem is not convex in both A and X rather either convex in A or X. Therefore the alternating update rule is suitable to find the optimal NTF representation. In [8, 9], it is also shown that the conventional additive update procedure can be transformed into a multiplicative update rule.

The multiplicative alternating update rules obtained by applying the standard gradient descent technique to the cost function shown in Eq.(2.10) is given by Eq. (2.12a) and Eq.(2.12b).

$$a_{ij} \leftarrow a_{ij} \frac{[XS^T]_{ij}}{[ASS^T]_{ij}}, \quad \textbf{(2.12a)}$$

$$s_{jk} \leftarrow s_{jk} \frac{\left[A^T X\right]_{jk}}{\left[A^T AS\right]_{jk}}. \qquad \textbf{(2.12b)}$$

This scalar form of update rules can be expressed in matrix notation as in Eq.(2.13)

$$A \leftarrow A \cdot\times XS^T \cdot / ASS^T \ , \qquad \textbf{(2.13a)}$$

$$S \leftarrow S \cdot\times A^T X \cdot / A^T AS, \qquad \textbf{(2.13b)}$$

where $\cdot\times$ and $\cdot /$ are componenet-wise mutiplication and division operators, respectively.

On the other hand, applying gradient descent technique to the KL-divergence given by Eq.(2.11) leads to the the multiplicative alternating update rules shown in Eq.(2.14),

$$a_{ij} \leftarrow a_{ij} \frac{\sum_{k=1}^{T} s_{jk} \left(\frac{x_{ik}}{[AS]_{ik}} \right)}{\sum_{p=1}^{T} s_{jp}}, \qquad \textbf{(2.14a)}$$

$$s_{jk} \leftarrow s_{jk} \frac{\sum_{i=1}^{I} a_{ij} \left(\frac{x_{ik}}{[AS]_{ik}} \right)}{\sum_{q=1}^{I} a_{pj}}. \qquad \textbf{(2.14b)}$$

Several regularized versions of the optimization described above are proposed recently. In [16], the regularized cost functions are expressed as,

$$D_F^{\alpha_A,\alpha_S}(X \,||\, AS) = \frac{1}{2}\left\|X - AS\right\|_F^2 + \alpha_A J_A(A) + \alpha_S J_S(S), \qquad \textbf{(2.15a)}$$

$$D_{KL}^{\alpha_A,\alpha_S}(X \,||\, AS) = \sum_{ik}\left(x_{ik} \log \frac{x_{ik}}{[AX]_{ik}} + [AX]_{ik} - x_{ik} \right) + \alpha_A J_A(A) + \alpha_S J_S(S), \qquad \textbf{(2.15b)}$$

s.t. $\forall i,j,k : s_{jk} \geq 0, a_{ij} \geq 0$, where α_A and α_S are regularization parameters and functions $J_A(A)$ and $J_S(S)$ are used to enforce certain application-dependent characteristics of the solution, such as sparseness.

Applying standard gradient descent technique to Eq.(2.15a) leads to the generalized learning rules given by Eq.(2.16),

$$a_{ij} \leftarrow a_{ij} \frac{\left[\left[XS^T\right]_{ij} - \alpha_A \left[\nabla_A J_A(A)\right]_{ij}\right]_\varepsilon}{\left[ASS^T\right]_{ij}}, \tag{2.16a}$$

$$s_{jk} \leftarrow s_{jk} \frac{\left[\left[A^T X\right]_{jk} - \alpha_S \left[\nabla_S J_S(S)\right]_{jk}\right]_\varepsilon}{\left[A^T AS\right]_{jk}}, \tag{2.16b}$$

where the operator $[y]_\varepsilon$ can be defined as $\max\{\varepsilon, y\}$ with small ε, i.e. projection on nonnegative orthant. Typically, ε is chosen as equal to 10^{-16} [20]. For sparse representations, the cost functions $J_A(A)$ and $J_S(S)$ can be selected as,

$$J_A(A) = \sum_{i=1}^{I} \sum_{j=1}^{R} a_{ij}, \tag{2.17a}$$

$$J_S(S) = \sum_{j=1}^{R} \sum_{k=1}^{T} s_{jk}, \tag{2.17b}$$

which correspond to the L_1-norms of matrices A and S, respectively. Therefore the multiplicative alternating update rules given by Eq.(2.16) are simplified to,

$$a_{ij} \leftarrow a_{ij} \frac{\left[\left[XS^T\right]_{ij} - \alpha_A\right]_\varepsilon}{\left[ASS^T\right]_{ij}}, \tag{2.18a}$$

$$s_{jk} \leftarrow s_{jk} \frac{\left[\left[A^T X\right]_{jk} - \alpha_S\right]_\varepsilon}{\left[A^T AS\right]_{jk}}. \tag{2.18b}$$

The sparseness control is ensured by normalizing the columns of A in each iteration. Also, it has been found that performance of the algorithms can be improved by this normalization. The column normalization of A can be expressed as in Eq.(2.19),

$$a_{ij} = \frac{a_{ij}}{\sum_{j=1}^{R} a_{ij}}. \qquad \textbf{(2.19)}$$

2.3.2 Generalized β-divergence

The β-divergence was first proposed by Minami and Eguchi for application in BSS [22], and the generalized divergence that unified the Euclidean distance and the Kullback-Leibler divergence was proposed by Kompass[23]. In this section of the thesis, update rules which are built on this divergence, proposed by Kompas, are stated.

The general form of cost function of β-divergence is given by Eq.(2.20),

$$D^{\beta}(X, AS) = \sum_{i=1}^{I}\sum_{k=1}^{T} \begin{cases} x_{ik}\dfrac{x_{ik}^{\beta} - [AS]_{ik}^{\beta-1}}{\beta(\beta+1)} + \dfrac{1}{\beta+1}[AS]_{ik}^{\beta}[AS - X]_{ik}, \beta \in [-1,1] \\ x_{ik}\left(\log x_{ik} - \log[AS]_{ik}\right) + [AS]_{ik} - x_{ik}, \beta = 0 \end{cases}. \qquad \textbf{(2.20)}$$

The Kompas update rules can be expressed as in Eq.(2.21),

$$a_{ij} \leftarrow a_{ij} \frac{\sum_{k=1}^{T} s_{jk}[AS]_{ik}^{\beta-1}}{\sum_{p=1}^{T} s_{jp}[AS]_{ik}^{\beta}}, \qquad \textbf{(2.21a)}$$

$$s_{jk} \leftarrow s_{jk} \frac{\sum_{i=1}^{I} a_{ij}[AS]_{ik}^{\beta-1}}{\sum_{q=1}^{I} a_{qj}[AS]_{ik}^{\beta}}. \qquad \textbf{(2.21b)}$$

If the same regularization terms are merged into Eq.(2.20), as in Eq.(2.15a) and (2.15b), the update rules can be written as in Eq.(2.22),

$$a_{ij} \leftarrow a_{ij} \frac{\left[\sum_{k=1}^{T} s_{jk} [AS]_{ik}^{\beta-1} - \alpha_A\right]_{\varepsilon}}{\sum_{p=1}^{T} s_{jp} [AS]_{ik}^{\beta}}, \quad \textbf{(2.22a)}$$

$$s_{jk} \leftarrow s_{jk} \frac{\left[\sum_{i=1}^{I} a_{ij} [AS]_{ik}^{\beta-1} - \alpha_S\right]_{\varepsilon}}{\sum_{q=1}^{I} a_{qj} [AS]_{ik}^{\beta}}. \quad \textbf{(2.22b)}$$

It is important to note that for β=1 the squared Euclidean distance, expressed by Frobenius norm, is obtained. On the other hand for the singular cases β =0 and β =-1, once the limits are evaluated KL-divergence and dual KL-divergence are obtained, respectively[22].

The choice of the parameter β depends on the statistical distribution of the data [22]. For example, the optimal choice of the parameter for the normal distribution is β=1, for the γ-distribution is β=-1, and β→0 for the Poisson distribution [9]. In the special case of β=1, a new algorithm called fixed point alternating least squares (FPALS), is proposed [20]. The FPALS update rules are expressed in Eq.(2.23)

$$A \leftarrow \left[\left(XS^T - \alpha_A E_A\right)\left(SS^T + \gamma_S E\right)^+\right]_{\varepsilon}, \quad \textbf{(2.23a)}$$

$$S \leftarrow \left[\left(A^T A + \gamma_A E\right)^+ \left(A^T X - \alpha_S E_S\right)\right]_{\varepsilon}, \quad \textbf{(2.23b)}$$

where γ_A, and γ_S are small nonnegative regularization coefficients, A^+ denotes Moore-Penrose pseudo-inverse of A and $E_A \in \Re^{I \times R}$, $E_S \in \Re^{R \times T}$, $E \in \Re^{R \times R}$ are matrices with all entries equal to one.

2.3.3 Generalized α-divergence

The algorithms derived for both NMF and NTF, are generally obtained by employing on of the three large classes of generalized divergences: the Bregman divergences, Amari's alpha divergence[14, 17, 20, 24], and Csisz´ar's φ-divergences. Here, the

update rules which are derived from the Amari's alpha divergence are stated. The generalized form of Amari's alpha divergence is shown in Eq. (2.24)

$$D_A^{(\alpha)}\left(x_{ik} \,||\, [AS]_{ik}\right) = \sum_{ik} \frac{x_{ik}^{\alpha}[AS]_{ik}^{1-\alpha} - \alpha x_{ik} + (\alpha-1)[AS]_{ik}}{\alpha(\alpha-1)}. \quad \textbf{(2.24)}$$

It should be noted that for α = 2, 0.5, and -1 the expression given by Eq.(2.24) turns into Peason's, Hellinger's and Neyman's chi-square distances, respectively. Also for the limit values of α (α → 1 and α → 0), the generalized KL-divergence and dual generalized KL-divergence are obtained, respectively. It is proposed that [16] instead of applying the standard gradient descent method, a nonlinearly transformed gradient approach can be used to derive the update rules which are given by Eq.(2.25),

$$\Phi(a_{ir}) \leftarrow \Phi(a_{ir}) - \eta_A \frac{\partial D_A^{(\alpha)}(X \,||\, AS)}{\partial \Phi(a_{ir})}, \quad \textbf{(2.25a)}$$

$$\Phi(s_{rt}) \leftarrow \Phi(s_{rt}) - \eta_S \frac{\partial D_A^{(\alpha)}(X \,||\, AS)}{\partial \Phi(s_{rt})}, \quad \textbf{(2.25b)}$$

where $\Phi(x) = x^{\alpha}$. Therefore by evaluating the derivatives and selecting the step sizes properly, the update rules can be written as in Eq.(2.26),

$$a_{ir} \leftarrow a_{ir}\left(\frac{\sum_{t=1}^{T} s_{rt}\left(x_{it}/[AS]_{it}\right)^{\alpha}}{\sum_{t=1}^{T} s_{rt}}\right)^{1/\alpha}, \quad \textbf{(2.26a)}$$

$$s_{rt} \leftarrow s_{rt}\left(\frac{\sum_{p=1}^{I} a_{pr}\left(x_{pt}/[AS]_{pt}\right)^{\alpha}}{\sum_{p=1}^{I} a_{pr}}\right)^{1/\alpha}. \quad \textbf{(2.26b)}$$

2.3.4 Alternating Least Squares Algorithm

Alternating least squares algorithm is another well-know approach that can be used for matrix factorization. ALS algorithm exploits the fact that, while the optimization is not convex in both A and S, it is convex in either A or S. Therefore, given one matrix the other can be found with simple least square computation [13, 21]. The simpliest form of ALS algorithm can be derived by first evaluating the gradient of square Euclidean distance in Eq.(2.10) with respect to S and A, then solving the expression, obtained by equalizing the gradients to zero, for S and A alternatingly. The gradients of Eq.(2.10) with respect to S and A are given by Eq.(2.27),

$$\nabla_S D_F(X \,||\, AS) = A^T AS - A^T X, \tag{2.27a}$$

$$\nabla_A D_F(X \,||\, AS) = ASS^T - XS^T. \tag{2.27b}$$

By equalizing Eq.(2.27a) and Eq.(2.27b) to zero and solving for S and A respectively, the following update rules are obtained,

$$S = (A^T A)^{-1} A^T X, \tag{2.28a}$$

$$A = XS^T (SS^T)^{-1}. \tag{2.28b}$$

However non-negativity condition is not satisfied in update rules given by Eq.(2.28). Simpliest approach to overcome this problem is to employ projection on non-negative orthant, hence the ALS update rules take the following form,

$$S \leftarrow \max\{\varepsilon, (A^T A)^{-1} A^T X\}, \tag{2.29a}$$

$$A \leftarrow \max\{\varepsilon, XS^T (SS^T)^{-1}\}. \tag{2.29b}$$

The ALS algorithm, like all the other algorithms explained before, can be regularized and enforced to be sparse. It is proposed by Cichocki and Zdunek that a more general

and flexible cost function, with regularization and sparsity penalties, can be employed [20].

The cost function, proposed by Cichocki and Zdunek[20], is given by Eq.(2.30),

$$D_F^{(\alpha)}(X \,||\, AS) = \frac{1}{2}\left\|W^{-1/2}(X - AS)\right\|_F^2 + \alpha_{As}\|A\|_{L1} + \alpha_{Ss}\|S\|_{L1} \qquad \textbf{(2.30)}$$

$$+ \alpha_{Ar}\left\|W^{-1/2}AL_A\right\|_F^2 + \frac{\alpha_{Sr}}{2}\|L_S S\|_F^2,$$

where $W \in \Re^{I\times I}$ is symmetric, positive definite, weighting matrix, $\alpha_{As} \geq 0$ and $\alpha_{Ss} \geq 0$ are parameters controlling a sparsity level of the matrices, and $\alpha_{Ar} \geq 0$, $\alpha_{Sr} \geq 0$ are regularization coefficients. The penalty terms $\|A\|_{L1}$ and $\|S\|_{L1}$ are the L_1 norms enforcing sparseness of A and S, respectively. L_A and L_S are regularization matrices which are selected according to characteristics of the application[5].

By evaluating the gradients of the cost function given by Eq.(2.30) with respect to A and S, Eq.(2.31) is obtained.

$$\frac{\partial D_F^{(\alpha)}(X \,||\, AS)}{\partial A} = W^{-1}(AS - X)X^T + \alpha_{As}E_A + \alpha_{Ar}W^{-1}AL_A L_A^T, \qquad \textbf{(2.31a)}$$

$$\frac{\partial D_F^{(\alpha)}(X \,||\, AS)}{\partial X} = A^T W^{-1}(AS - X) + \alpha_{Ss}E_S + \alpha_{Sr}L_S^T L_S S, \qquad \textbf{(2.31b)}$$

where the size of E_A is equal to size of A and with all elements equal to one, E_S is a matrix of size equal to size of S and with all elements equal to one. By equalizing Eq.(2.31a) and Eq.(2.31b) to zero and solving for A and S, the following update rules are obtained, respectively:

[5] L_A and L_S are chosen as unit diagonal matrices throughout this work.

$$A \leftarrow (XS^T - \alpha_{As} WE_A)(SS^T + \alpha_{Ar} L_A L_A^T)^{-1}, \quad \textbf{(2.32a)}$$

$$X \leftarrow (A^T W^{-1} A + \alpha_{Sr} L_S^T L_S)^{-1} (A^T W^{-1} X - \alpha_{Ss} E_S). \quad \textbf{(2.32b)}$$

The update rule shown in Eq.(2.32) is named as fixed point regularized alternating least squares by Cichocki and Zdunek [20]. It can be considered as a generalized version of the ALS algorithms, since it simplifies to standard ALS when all the regularization parameters are set to zero.

It is proposed that the separation performance of the algorithms can be improved by changing the parameters dynamically depending on the iteration index, rather than fixing them throughout the iterations[20]. This can be expressed as in Eq.(2.33),

$$\alpha(k) = \alpha_0 \exp\{-k / \tau\}, \quad \textbf{(2.33)}$$

where $\alpha_0 = \alpha(0)$ is the initial value of α, k is the iteration index, τ is the step size. The selection of α and τ is problem dependent, therefore they can only be set experimentally.

3. DESIGNED SYSTEM AND TEST RESULTS

In this section of the thesis, the software which is designed to realize the factorization algorithms is explained in details. Using this software separation performances of the algorithms, explained in the previous section, are tested under several different conditions. Subsection 3.1 presents the developed sytem and designed graphical user interface. Test cases and experimental results obtained on a number of audio sources are reported in subsection 3.2

3.1 GUI and Designed System

The software is developed in JAVA environment and it consists of three main modules; input/preprocessing, mixing, reconstruction. Characteristic of each module is described in the following subsections.

3.1.1 Preprocessing Module

Input of the software is digital sound files in wave(wav) format. These sound files are browsed and selected and the software loads the source signals after performing preprocessing. The loaded source signals(audio files) can be ploted and played by using the designed software. These input sound files contain the original source signals which are supposed to be accurately reconstructed after mixing.

The nonnegative tensor factorization states that the source signals must be non-negative to acquire successful separation, however audio(source) signals have negative values, inherently. Therefore a simple preprocessing operation is performed to make audio signals positive. The process of making source signals positive is performed on each source signal, separately. At the preprocessing step, the minimum non-negative value of the source signal is calculated and its absolute value is added to each sample of the source signal. This can simply be stated as, shifting the amplitude of the source signal up. The graphical user interface of the desinged

software with two input signals is depicted in Fig.3.1. Here, it can be seen that amplitude of source signal samples are all positive.

Refering back to mathematical representation of the NTF, these input signals are the the columns of the matrix $S \in \Re^{RxT}$, given in Eq.(2.3). The number of rows R, is the number of source signals and the number of columns T, is the number of samples in the source signals. If the number of samples in the source signals is not equal, meaning that lenght of audio files are different, then the source signals with smaller lenght are padded with zeros to equalize the lenghts. The zero paddings in the waveform of the first source signal, plotted on the top, can be clearly seen in Fig.3.1.

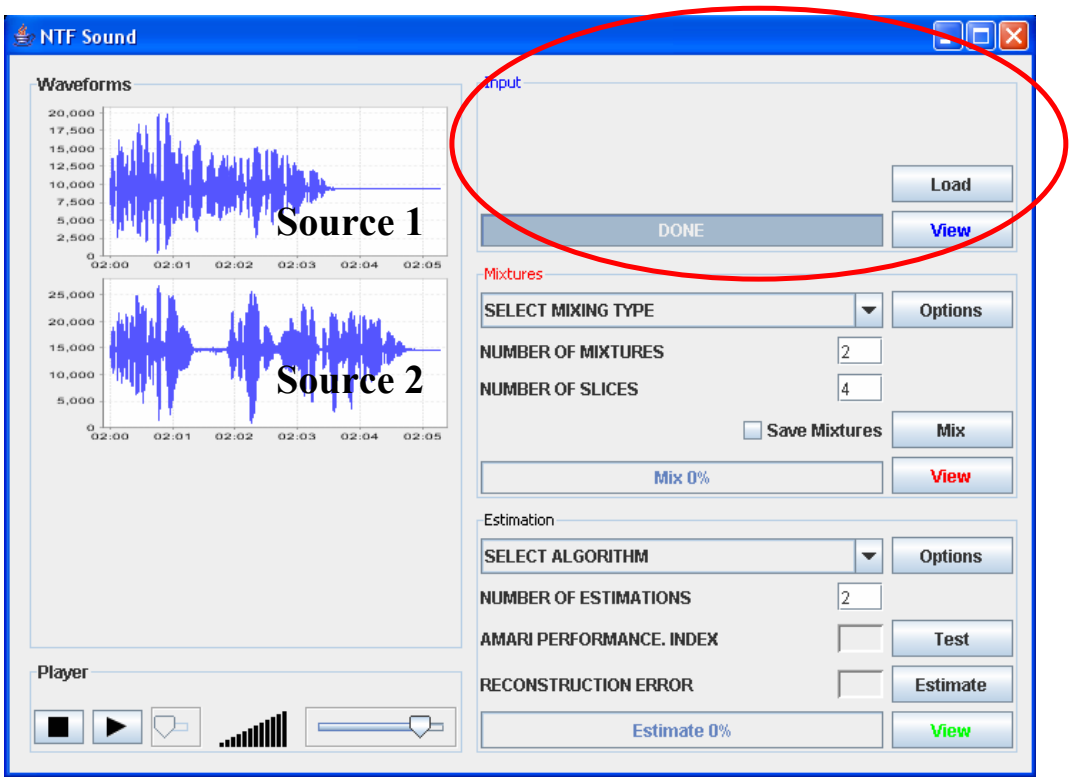

Figure 3.1 : The graphical user interface of the designed software with two sound files loaded and displayed.

3.1.2 Mixture Generation Module

Once the source signals are loaded, the next step is to create mixtures which are assumed to be the linear combinations of the source signals. The mixtures can be generated in three different ways;

- Using another sound processing software(third-party product).
- Directly recording by microphones, i.e.: Acoustic mixtures.
- Using the designed software.

The first two ways, listed above, are out of the control of the designed software. In those ways, the mixtures are created externally and then loaded into the software. However, mixtures can also be generated by using the software itself. The advantage of generating the mixtures within the software is that by this way different mixing conditions can be evaluated. This is accomplished by first defining the mixing matrix A and then multiplying the source matrix S which contains the input(source) signals, as it is given in Eq.(2.3). Result of this matrix multiplication yields the mixtures, X. This is the general way of generating mixtures in the designed software, to be more specific about the matrix structures and dimensions, suppose the 3D-NTF2 model, given in Fig.3.2, is used.

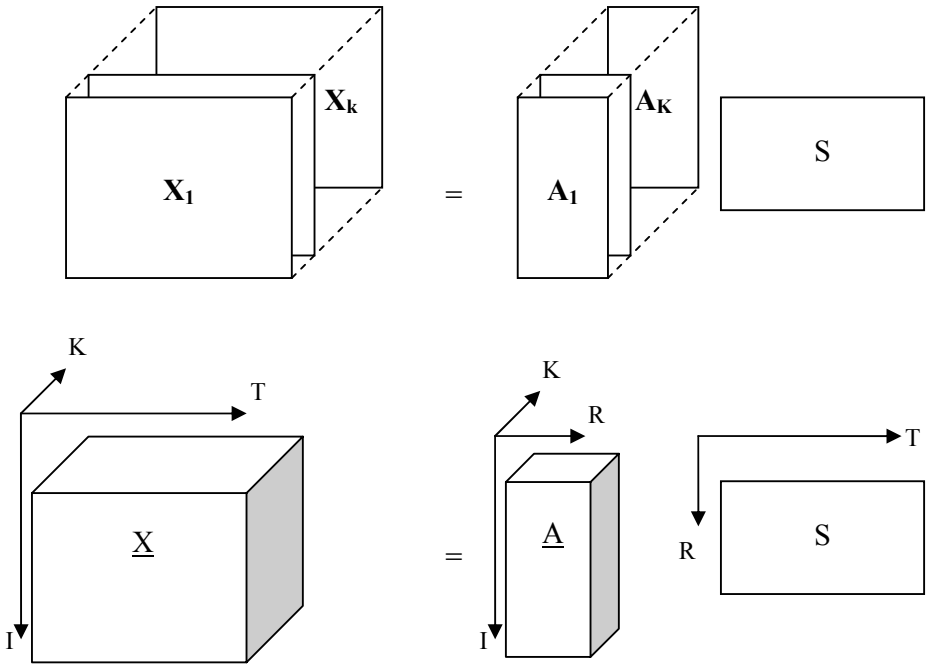

Figure 3.2 : Simplified 3D-NTF2 Model

$$X_k = A_k S, (k = 1, 2,...,K) \quad \textbf{(3.1)}$$

The mathematical expression of this model is given in Eq.(3.1), where $X_k \in \Re_+^{I \times T}$ is the frontal slice of the tensor $\underline{X}$ containing the audio mixtures, $A_k \in \Re_+^{I \times R}$ is the

frontal slice of the mixing tensor (i.e.: each A_k that constitutes $\underline{A}$ is a mixing matrix.) and $S \in \Re_{+}^{R \times T}$ is the matrix which contains the original source signals in its rows. Here, I is the number of mixtures, T is the number samples in audio signals, R is the number of source signals and K is the number of frontal slices. For the 3D-NTF2 model we can also define K as the number of mixing matrix, used to construct mixtures.

It is clear that, the tensor $\underline{X}$ which contains the mixtures is created slice by slice, by multiplying the source matrix S with each frontal slice of the tensor $\underline{A}$. Note that each frontal slice A_k of the tensor $\underline{A}$ corresponds to a different mixing matrix. Using this software, entries of each mixing matrix that mades up the mixing tensor $\underline{A}$ can be set by the user. Throughout this work the entries of mixing matrices are chosen among uniformly distributed random numbers, however there are other options implemented in the software.

3.1.3 Estimation of Source Signals via Reconstruction Module

This part of the software is designed to fullfil the blind audio source separation task. Three different algorithms are implemeted within the software to estimate the source signals, and these algorithms are based upon the optimization techniques stated in the previous section of the thesis.

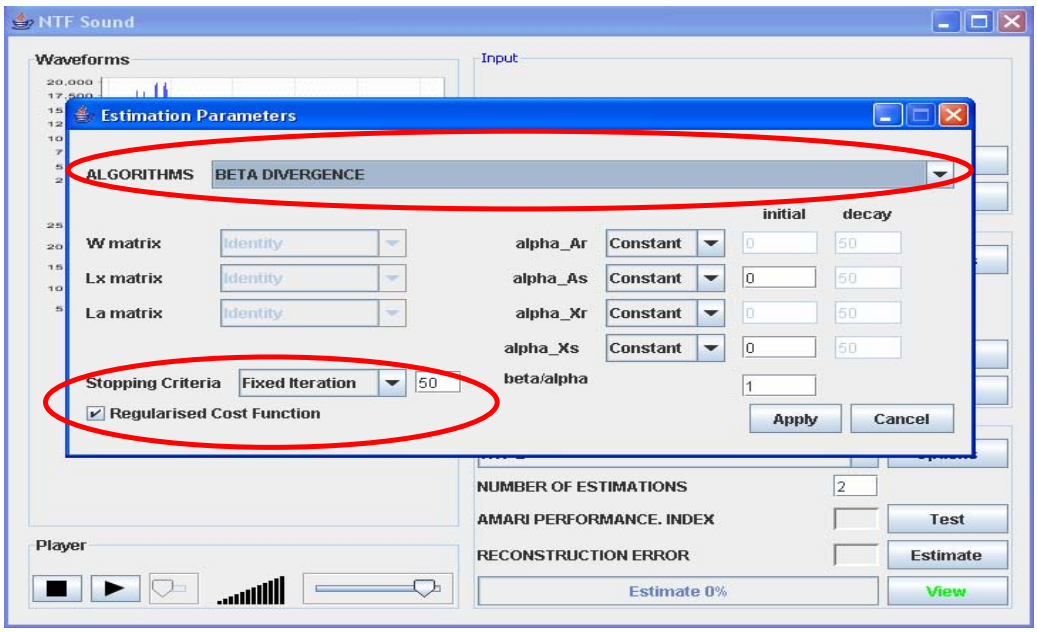

Figure 3.3 : The estimation parameters window of the interface.

Algorithm 1 : Fixed Point Alternating Least Squares (FPALS)

Initialize : W, La, Ls, α_{As}, $\alpha_{Ar,}$ $\alpha_{Xs,}$ α_{Xr}
Initialize : A (uniformly distributed random numbers)
*While (*condition*) do:*
    update S using Eq.(2.32b)
    set S = max{0, S}
    update A using Eq.(2.32a)
    set A = max{0, A}
    normalize the columns of A
end

Algorithm 2 : α-divergence

Initialize : A, S, α
*while(*condition*) do:*
    update S using Eq.(2.26b)
    set S = max{0, S}
    update A using Eq.(2.26a)
    set A = max{0, A}
    normalize the columns of A
e*nd*

Algorithm 3 : β-divergence

Initialize : A, S, β, α_A, α_S
*while(*condition*) do:*
    update S using Eq.(2.21b)
    set S = max{0, S}
    update A using Eq.(2.21a)
    set A = max{0, A}
    normalize the columns of A
end

As it is seen in Fig.3.3, the algorithm and corresponding parameters can be set using the estimation parameters window of the designed interface. One of the important parameter that must be set before starting the estimation is, the stopping criterion which specifies the condition in which the estimation ends. The stopping criterion parameter can either be set to fixed iteration number or to the estimation error obtained in each iteration, i.e: the estimation error is calculated in each iteration and

if the estimation error converges, the estimation terminates itself. Once all the parameters are set, the separation can be started. A snapshot of the interface after the estimation, is shown in Fig.3.4.

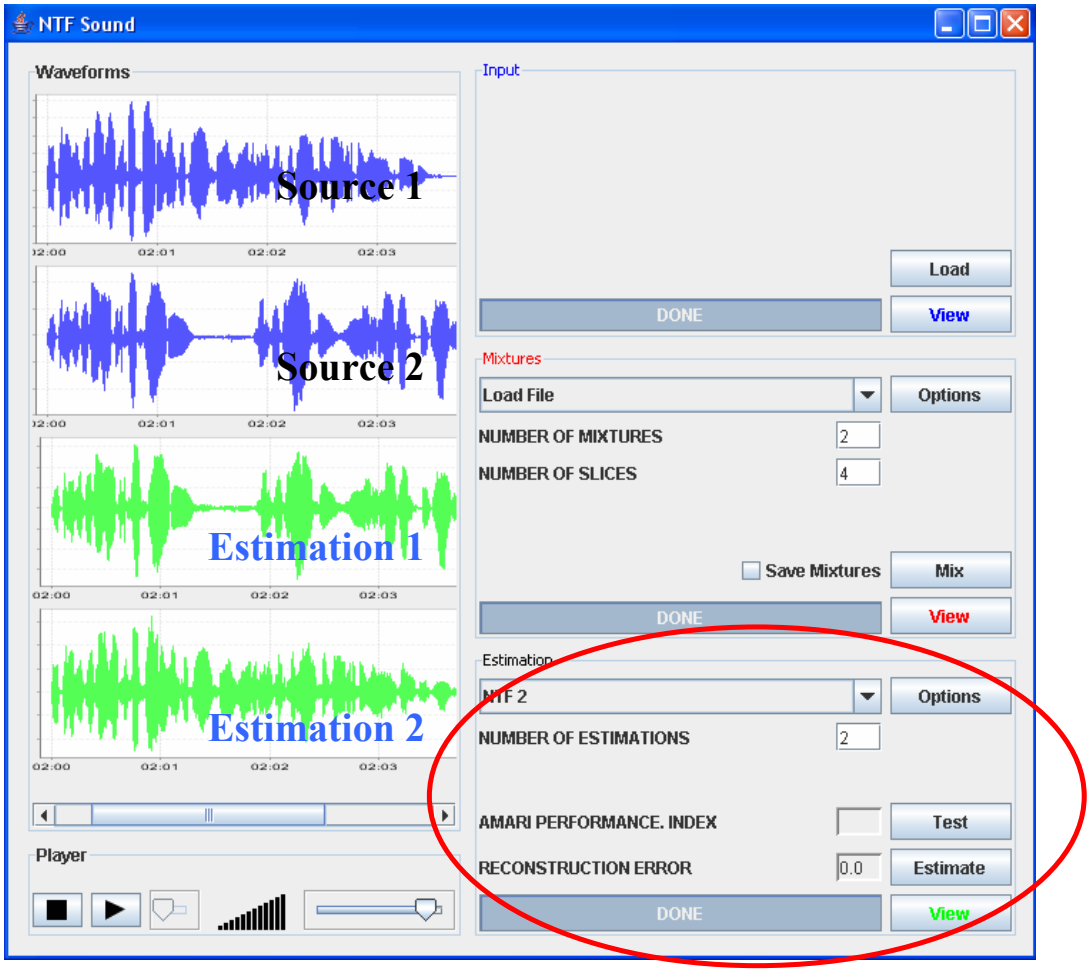

Figure 3.4 : The snapshot of the interface after the estimation is performed.

3.2 Test Cases and Results

The separation performance of the algorithms, explained before, are tested under several conditions. These test conditions can be classified as;

- the estimation performace of the algorithms compared to each other,
- the estimation performance of the algorithms on the noisy mixtures,
- the effect of initialization on the estimation performance,
- the effect of mixing matrix on the estimation performance,

- the performance of the algorithms on speech-speech mixtures and speech-audio mixtures,
- the effect of the number of slices on the estimation performance, i.e: the number of mixing matrices,
- the effect of regularization and sparseness terms on the estimation performance.

The audio files that are used in tests are all in wav format with 44100 sampling rate and 16 bit PCA encoding. These files consist of two groups; the audio files of the first group are the voices of the male and female speakers which are pronoucing the same phrase. Since each speaker pronounces the same phrase, the mixtures created using only these files can be considered as highly correlated. The second group of audio files are small samples of orchestra recordings which only contain instruments.

Two criterion is observed throughout the tests and the performance evaluations are based upon these two criterion. One of them is the norm of the reconstruction (estimation) error which is given in Eq.(3.2)

$$\left\|E\right\|_F = \left\|X - \hat{X}\right\|_F \tag{3.2}$$

where X represents the mixtures and $\hat{X}$ is the estimated mixtures obtained by multiplying the estimated mixing matrix $\hat{A}$ with the estimated source matrix $\hat{S}$. Since $\hat{X}$ is not directly estimated but reconstucted by using the estimations of source and mixing matrices, the error ($E = X - \hat{X}$) is also called the reconstruction error. By calculating the norm of the error matrix E after each update, the progress of the estimation can be observed. The other performance measure is the Amari performace index [25] which is given in Eq.(3.3).

$$P_{err} = \frac{1}{2N}\sum_{i,j=1}^{M}\left(\frac{\left|P_{ij}\right|}{\max_k\left|P_{ik}\right|} + \frac{\left|P_{ij}\right|}{\max_k\left|P_{kj}\right|}\right) - 1, \tag{3.3}$$

where $Pij = \left(\hat{A}^{-1}A\right)_{ij}$, A is the mixing matrix, $\hat{A}$ is the estimated mixing matrix[6]., M is the number of mixtures and N is the number of sources Througout this work the non-degenerative BSS case where there are at least as many mixtures as sources, is studied (i.e.: $M \geq N$). When this is the case, the accuracy of estimation can be evaluated from the accuracy of estimated mixing matrix. Therefore, Amari performance index is selected as a measure of estimation performance. Considering the degenerate BSS problem, the estimation performance can not be assessed only from the mixing matrix but also the estimation of sources should be taken into account. The most widely used index to measure the performance in degenerate BSS case is the Signal to Interference Ratio, (SIR) [25].

3.2.1 Evaluation of Estimation Performance of the Algorithms

The designed three algorithms are tested and their performances are compared to each other. In this subsection, results are reported. Tests are performed on five different set of mixtures generated randomly. The idea behind using randomly generated mixture matrices is to evaluate the performance under worst conditions. For this test we have generated 5 different mixing matrices given in Table A.1-5. Audio sources used for this test were Speech Source 1 and Speech Source 2 shown at top of Fig.3.1, and Fig.3.2, respectively. In Table 3.1, the Amari indices, calculated after runnning each of the BSS algorithms for fixed number of iterations, are listed. In the literature, it is common to accept the estimation having Amari index less than 0.03 is a good estimation. As it is seen in Table 3.1, The Alternating Least Squares (ALS) algorithm is superior to both of the other algorithms, beta divergence takes the second place and the alpha divergence shows the poorest performance. In these tests, the alpha value in alpha divergence is emprically set to 1, the beta value in beta divergence is emprically set to 1 with no regularization employed and also for the ALS algorithm neither sparseness nor regularization is applied. The best result of each algorithm is obtained for the third set of mixtures. Note that selection of the algorithm parameters is crucial for the performance of alpha and beta divergences. Therefore, more detailed tests are performed on the third set of mixtures by changing

[6] The matrix inversion is performed as Moore-Penrose pseudinverse operation, if $\hat{A}$ is not a square

the parameters of the algorithms. Fig.A.1 illustrates the mixed signals obtained by the third mixing matrix.

Table 3.1: Amari index of each estimation algorithm per mixture.

Mixture	ALPHA	BETA	ALS
1	0.142	0.095	0.092
2	0.178	0.216	0.084
3	0.15	0.075	0.039
4	0.213	0.166	0.055
5	0.092	0.084	0.053

Table 3.2: Amari index vs.different configuration of parameters.

Algorithm	Amari Index
α-divergence; α = 1	0.15
α-divergence; α = 2	0.098
α-divergence; α = 3	0.089
β-divergence; β =1, reg. const. = 0	0.075
β-divergence; β =1, reg. const. = 0.1	0.044
β-divergence; β =1, reg. const. = 0.5	0.056
β-divergence; β =1, reg. const. = 0.9	0.044
β-divergence; β =1, reg. const. = 100	0.036
β-divergence; β =1, reg. const. = 1000	0.034
ALS; no regularization	0.039

In Table 3.2, the Amari indices of estimations with different parameter configurations of the algorithms are reported. It is observed that the algorithms reach to the convergence after less than 100 iterations. Therefore, all the results reported in Tables are obtained after running the algorithms for 100 iterations. It is seen that, accuracy of the estimations which are achieved using alpha and beta divergences, can be improved by properly selecting the parameter values. Especially for the beta divergence, the choice of regularization constant radically effects the estimation performance and even gives better estimation results than ALS algorithm does, can be obtained. However from the view point of computational complexity, the run time of ALS algorithm is shorter than the other two. Despite the improved performance of the beta algorithm, the salient run time difference between beta and ALS algorithms is one of the most important drawback of beta divergence. Given these results, it can be generalized that ALS algorithm is superior compared to other two algorithms from both complexity and performance point of views.

---

matrix.

3.2.2 Effect of Initialization on the Estimation Performance

One of the important difference between NTF algorithms is the number of parameters initialized and the way of initializations made, before starting the estimations. In some of these algorithms, it is required that both S (source matrix) and A (mixing matrix) be initialized, on the other hand in some algorithms the initialization of either A or S is sufficient for algorithim to be run. The alpha and beta algorithms requires the initialization of both A and S whereas in ALS algorithm only A matrix is needed to be initialized. In all cases, it is known that the NTF algorithms are generally sensitive to the initialization and either the convergence speed or the local minimum obtained, can be improved with better initialization.

Almost all the NTF algorithms simply use random initiazation. Meaning that A and/or S is initialized as a dense matrices of random numbers. In most cases, it is assumed that the random initializataion is the worst but the simpliest way of initialization[26]. Here, the effect of random initialization on ALS algorithm is tested and the following results are obtained.

Table 3.3 : The effect of random initialization over the estimation performance.

Initialization	Reconstruction Error	Amari Index	Number of Iterations to Convergence
1	678.6829576	0.0395044	34
2	560.9257836	0.0384464	30
3	1632.072045	0.0395045	31
4	564.1097353	0.0300434	31
5	630.4005966	0.0395034	34

The ALS algorithm is run with 5 different random initialzations of mixing marix for 100 iterations. In Table 3.3, it is shown that the estmation performance is not effected considerably by using different random initialization. For all the five initializations, reconstruction errors, Amari indices and the number iterations to convergence are quite the same. It can be seen in Fig.3.5 and Fig.3.6 that the estimated waveforms does not differ a lot from each either.

As it is shown in Fig.3.7, the progress of estimations throughout the iterations is also very similar for all five initializations. From all these results, it can be deduced that the ALS algorithm is independent of the intializations made using uniformly distributed random numbers. However a better generalization can be derived by

comparing different initialization techniques. The effect of more elegant initialization techniques are left as a future work.

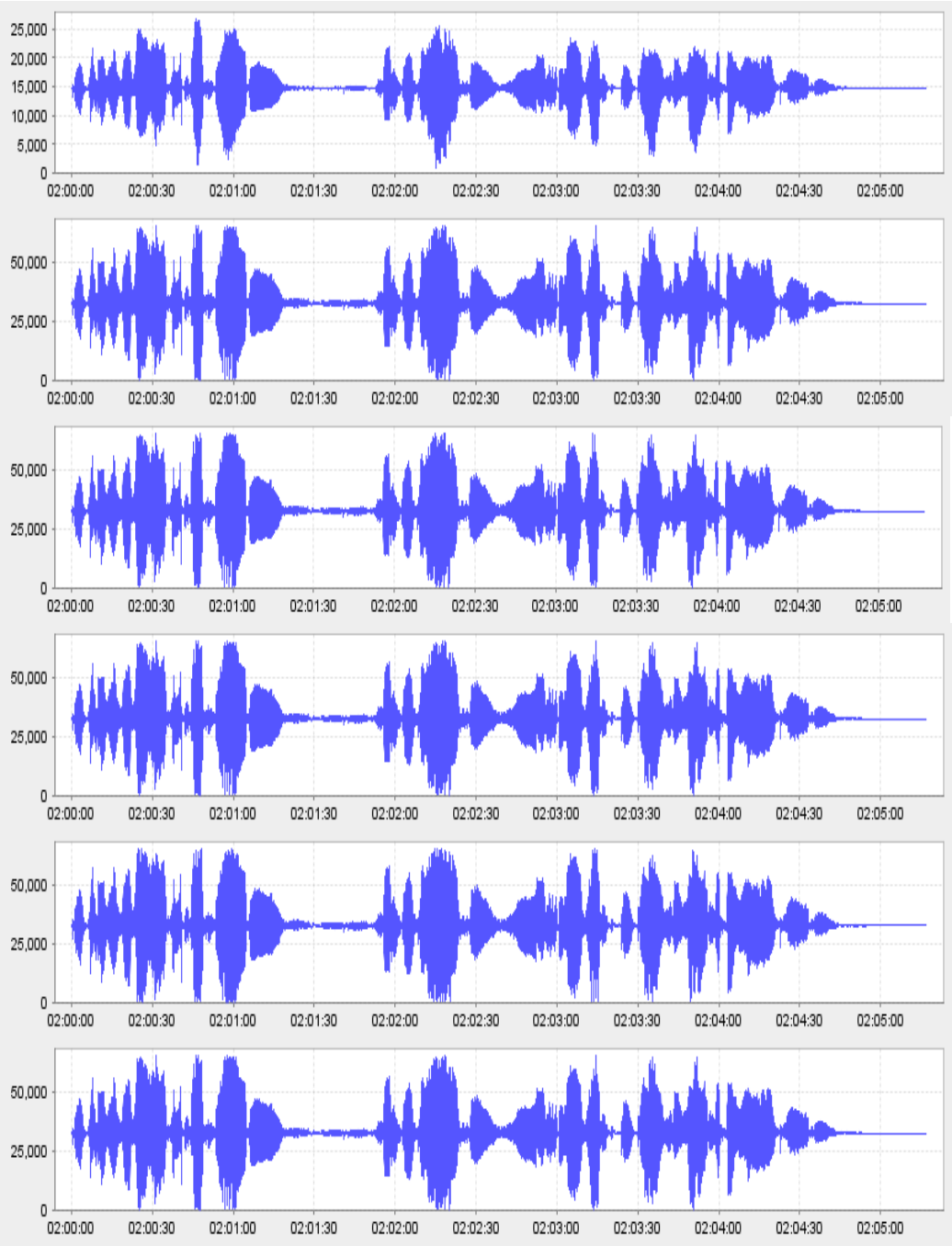

Figure 3.5: The estimations of first sources for each initialization. Top to bottom; Original Source 1, Estimated Source 1 with initialization 1, Estimated Source 1 with initialization 2, Estimated Source 1 with initialization 3, Estimated Source 1 with initialization 4, Estimated Source 1 with initialization 5.

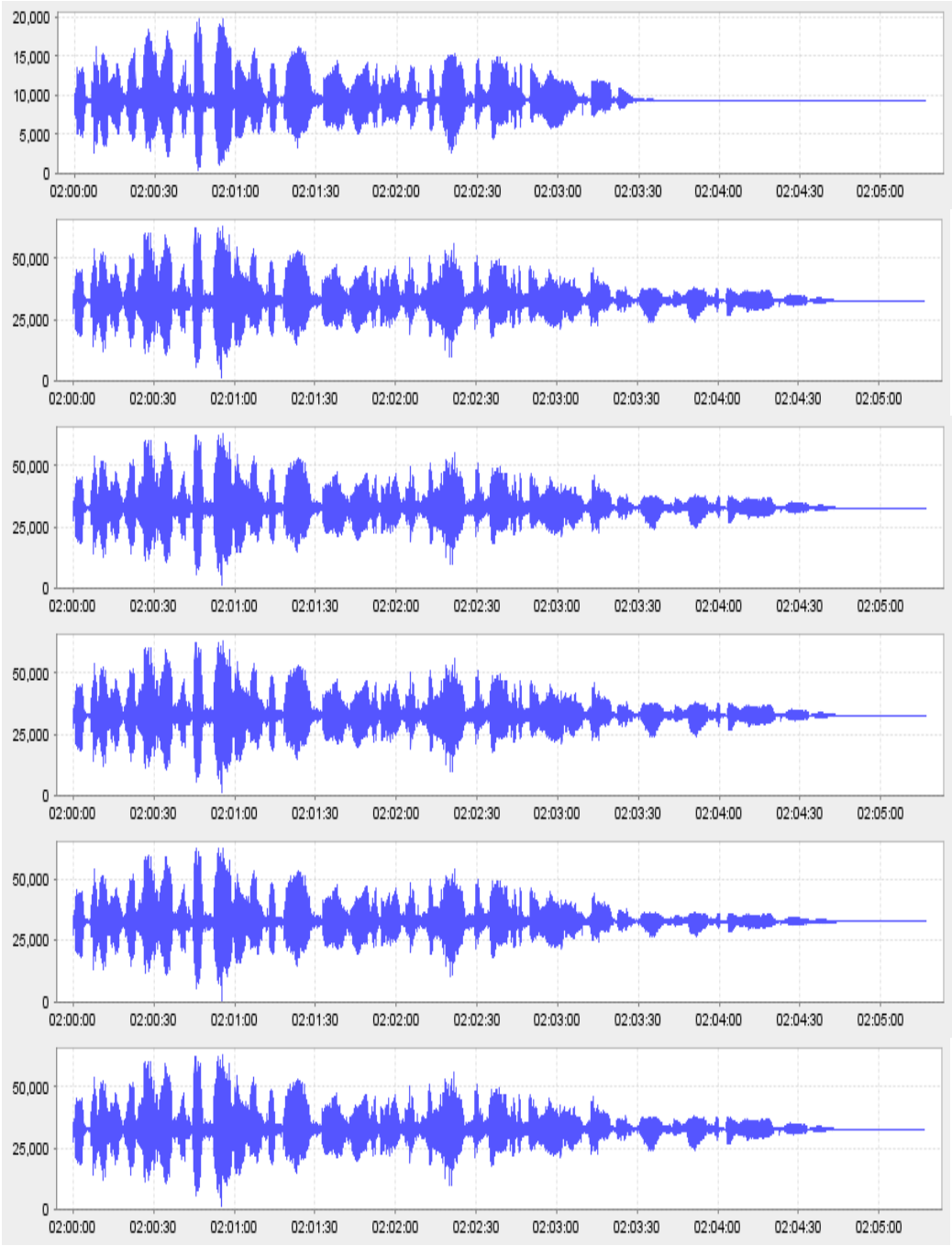

Figure 3.6: The estimations of second sources for each initialization. Top to bottom; Original Source 2, Estimated Source 2 with initialization 1, Estimated Source 2 with initialization 2, Estimated Source 2 with initialization 3, Estimated Source 2 with initialization 4, Estimated Source 2 with initialization 5.

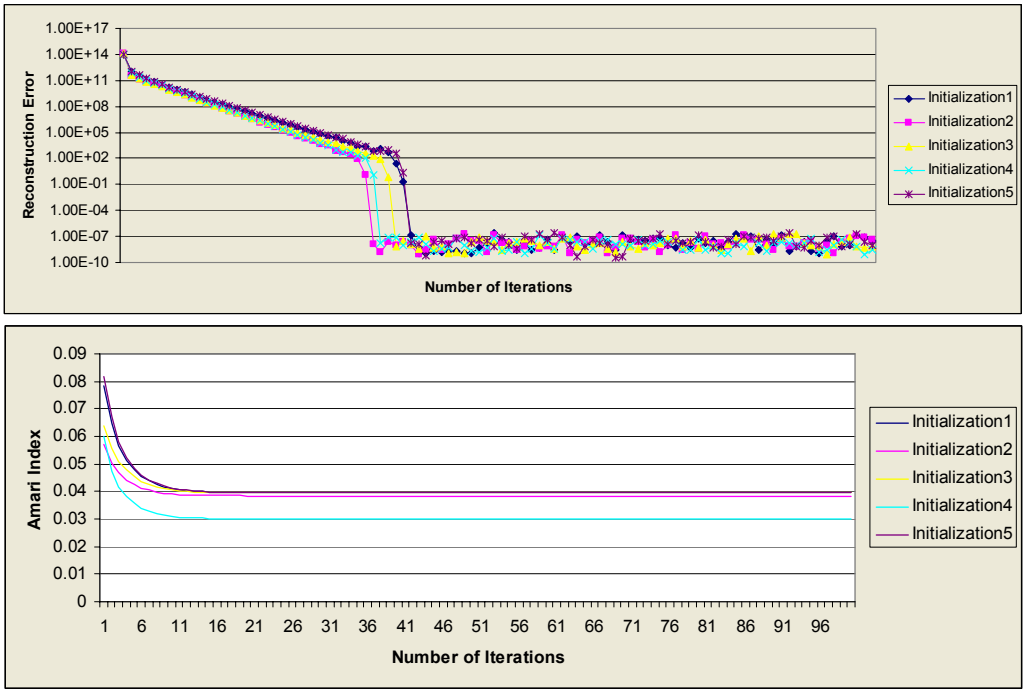

Figure 3.7: The change of reconstruction error (top) and Amari index (bottom) through the iterations for each initialization of A.

3.2.3 Effect of Mixing on the Estimation Performance

While evaluating the estimation performance the amount of mixing plays an important role. A poor algorithm can yield equally successful results when compared to a better algorithm, if the mixing is poor. Meaning that the mixtures are not mixed enough to make discrimination in between the algorithms. Here, the effect of mixing on ALS algorithm is investigated for different set of mixtures which are generated using randomly initialized mixing matrices and the mixing is perfomed linearly. The ALS algorithm is run wih 5 different set of mixtures for 100 iterations and the initializations are fixed for all cases. the results are given in Table 3.4.

Table 3.4 : The effect of mixing on the performance of the estimation.

Mixing	Reconstruction Error	Amari Index	Number of Iterations to Convergence
1	156.9628906	0.0926204	29
2	8.65E+09	0.0849004	100+
3	678.6829576	0.0395044	34
4	2752.05867	0.0555363	18
5	145.4309979	0.0536035	66

The effect of mixing matrix used, can be clearly seen in Table 3.4. Unlike the initialization case, for different mixing matrices, radically different local minima are obtained. The number of iterations required for the convergence is also different for each mixing matrix and for the second mixing, no convergence is achieved even after 100 iterations. The difference in the change of reconstruction error and Amari indices can also be observed in Fig. 3.8, The best Amari index is obtained in third mixing however, this does not satisfy neither the fastest convergence nor the best reconstruction error. This is due to the fact that the reconstruction error is calculated using both estimated source and mixing matrices, whereas the Amari index is obtained using only mixing matrix. The waveforms of the estimated sources are given in Fig.3.9 and Fig.3.10. The consistency between the estimated waveforms and the Amari indices are clear, however it should be noted that throughout the tests it is observed that waveforms can sometimes be misleading. Therefore to deduce a conclusion both the waveforms and the Amari indices should be taken into account.

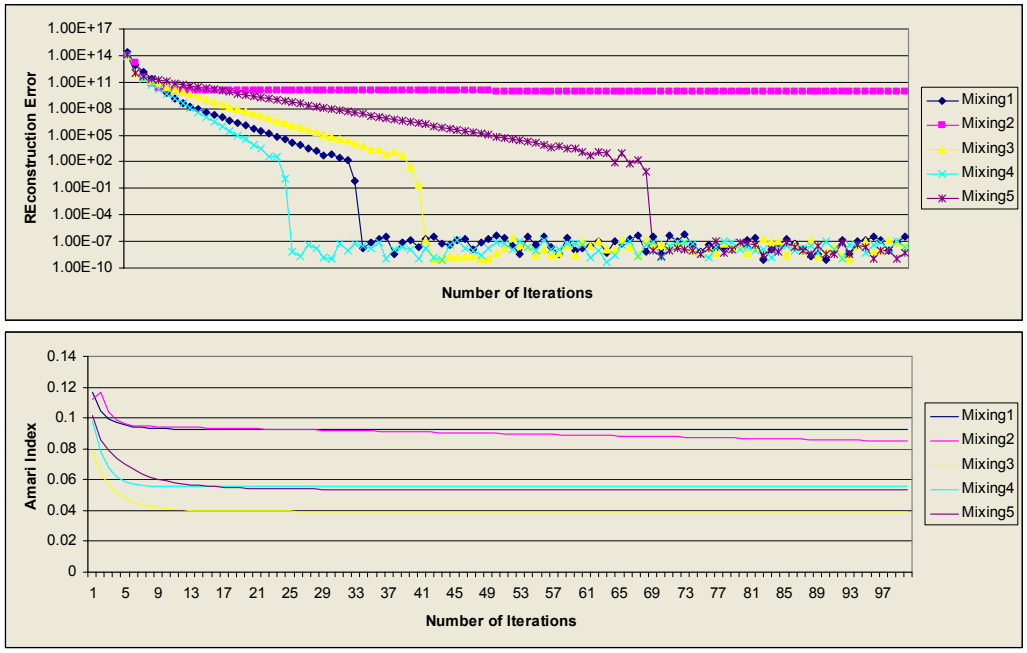

Figure 3.8: The change of reconstruction error (top) and Amari index (bottom) through the iterations for each mixing.

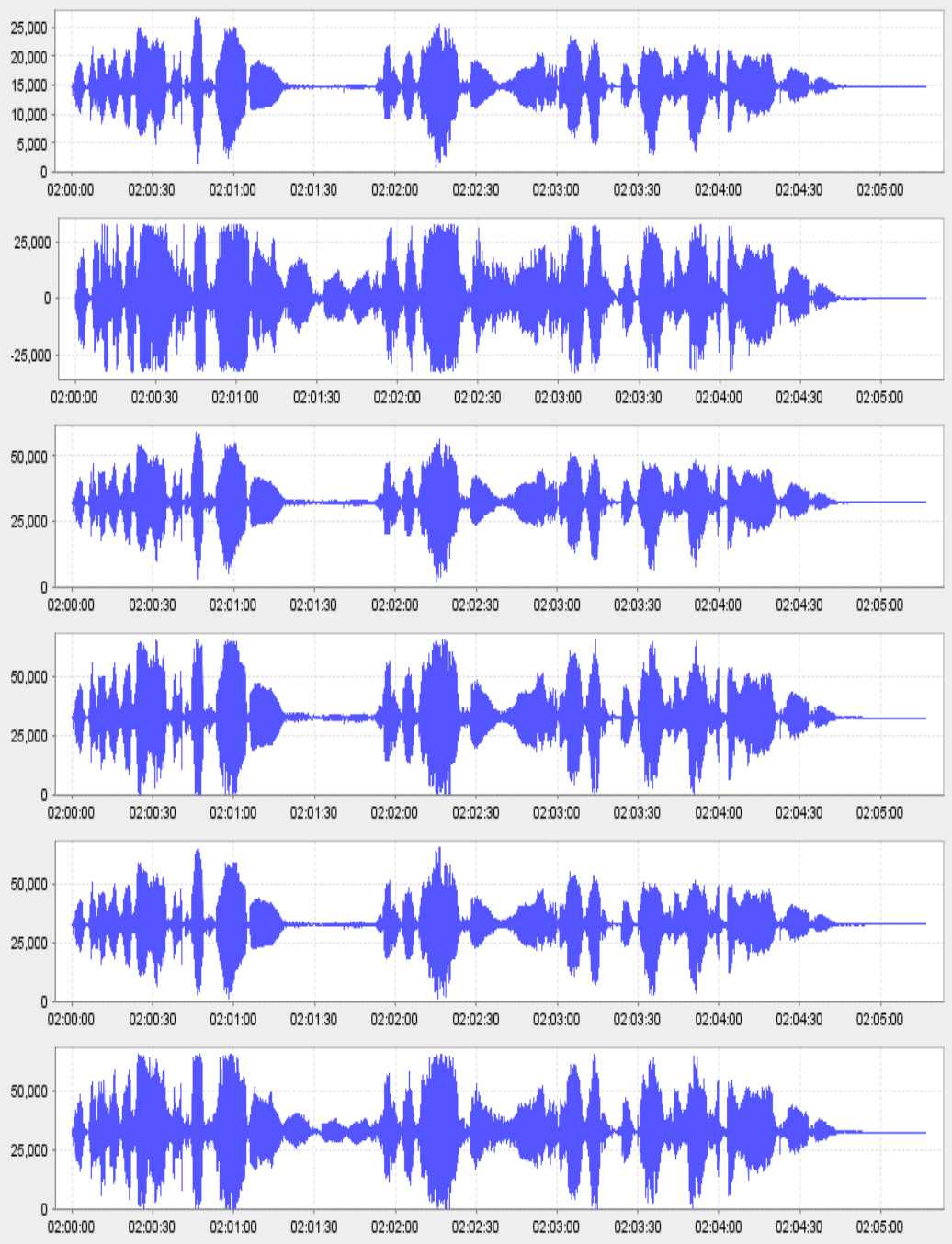

Figure 3.9: The estimations of first sources for each set of mixtures. Top to bottom; Original Source 1, Estimated Source 1 with mixing 1, Estimated Source 1 with mixing 2, Estimated Source 1 with mixing 3, Estimated Source 1 with mixing 4, Estimated Source 1 with mixing 5.

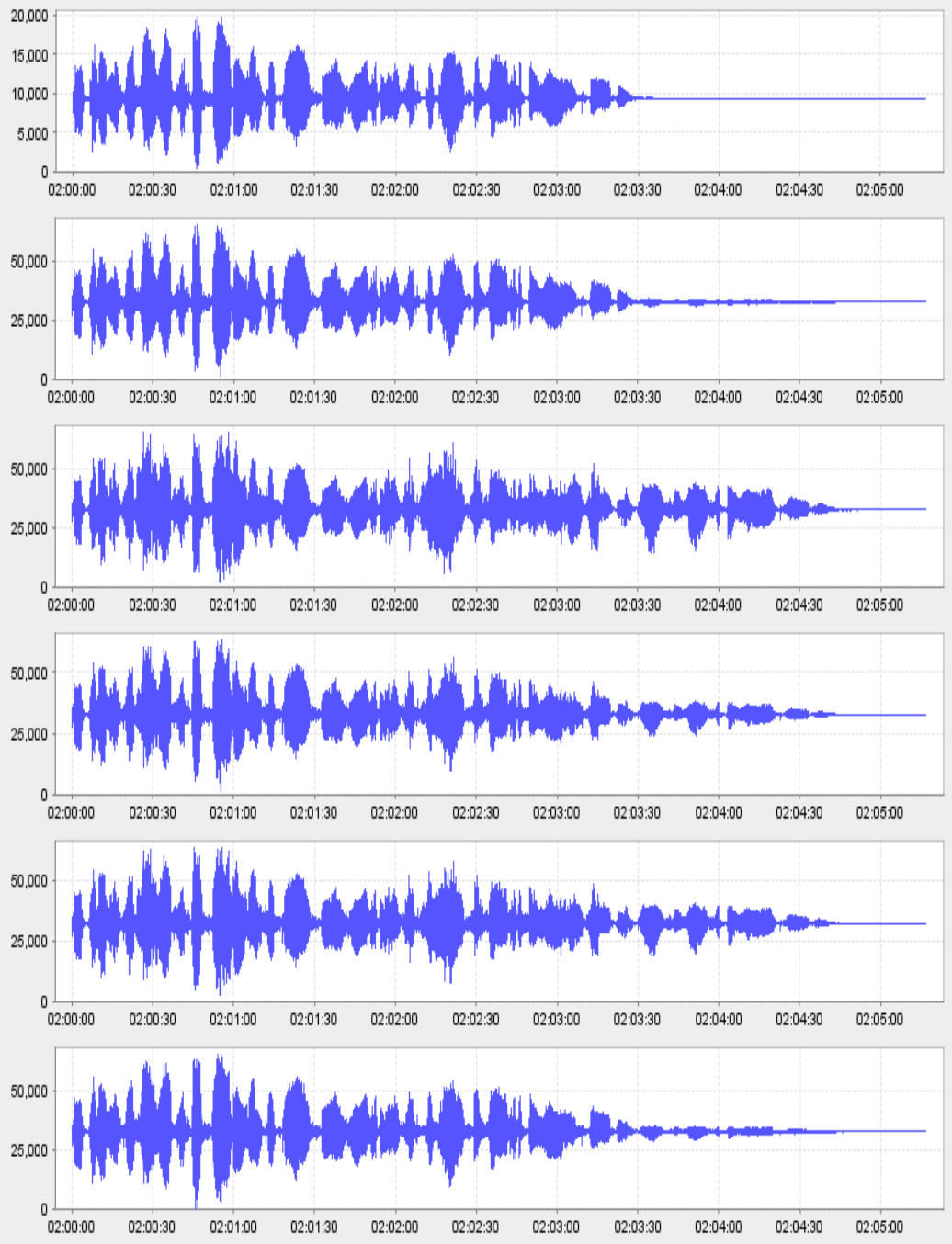

Figure 3.10: The estimations of second sources for each set of mixtures. Top to bottom; Original Source 2, Estimated Source 2 with mixing 1, Estimated Source 2 with mixing 2, Estimated Source 2 with mixing 3, Estimated Source 2 with mixing 4, Estimated Source 2 with mixing 5.

3.2.4 Effect of Slice Number K, on the Estimation Performance

The number of Slices K, is another important parameter therefore its effect is studied in detail. It is the slice number that actually makes the difference between NTF and NMF. The BSS problems which are solved with NTF approach can be solved with NMF approach, if the slice number is chosen as 1. In Table 3.5, the effect of slice number is given in terms of reconstruction error, amari index and the number of iterations.

Table 3.5 : The effect of number of slices on the estimation performance.

Number of Slices	Reconstruction Error	Amari Index	Number of Iterations to Convergence
1	2.18E+08	0.25983614	100
2	149.2798524	0.25091538	16
3	15.65047086	0.18391848	10
4	202.4943579	0.03950381	37

It is obvious that as the number of slices slice is increased, the better estimations are obtained. The increase in number of slice is actually nothing more than using more mixing matrices, hence more mixtures. Therefore as long as a the diversity can be increased by adding new mixing matrix, the better estimations can be obtained. On the other hand, since diversity is not guaranteed only by adding a new mixing matrix, the effect of increasing K can be negligible after a point. It should be noted that run time of the algorithms is also increases radically as the number of slice increase.

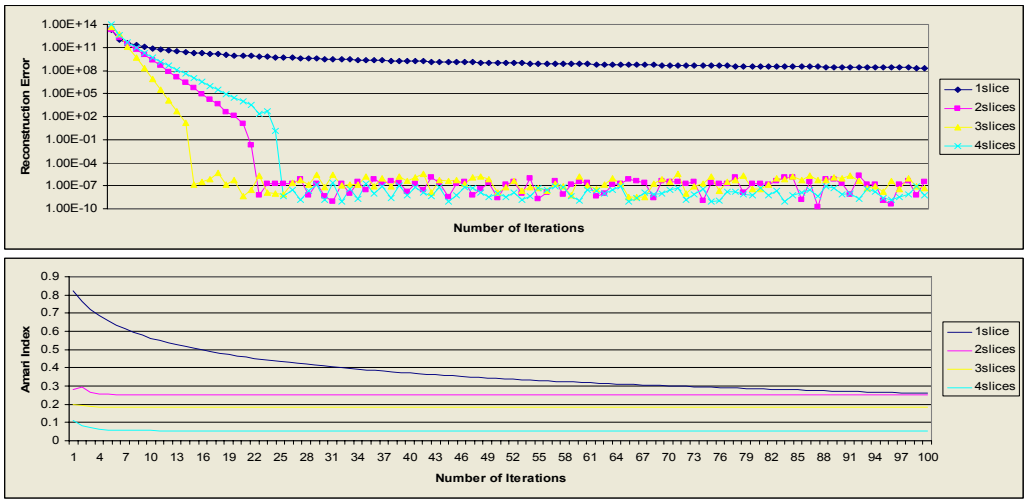

Figure 3.11: The change of reconstruction error (top) and Amari index (bottom) through the iterations for each number of slices.

The difference between the case K = 1 and the K = 4 can clearly be seen in Fig.3.11 in which the change of reconstruction error and Amari index is given. Also in Fig.3.12 and Fig.3.13, it can be seen that the waveforms estimated in K=4 case, are more similar to original sources than they are in other cases.

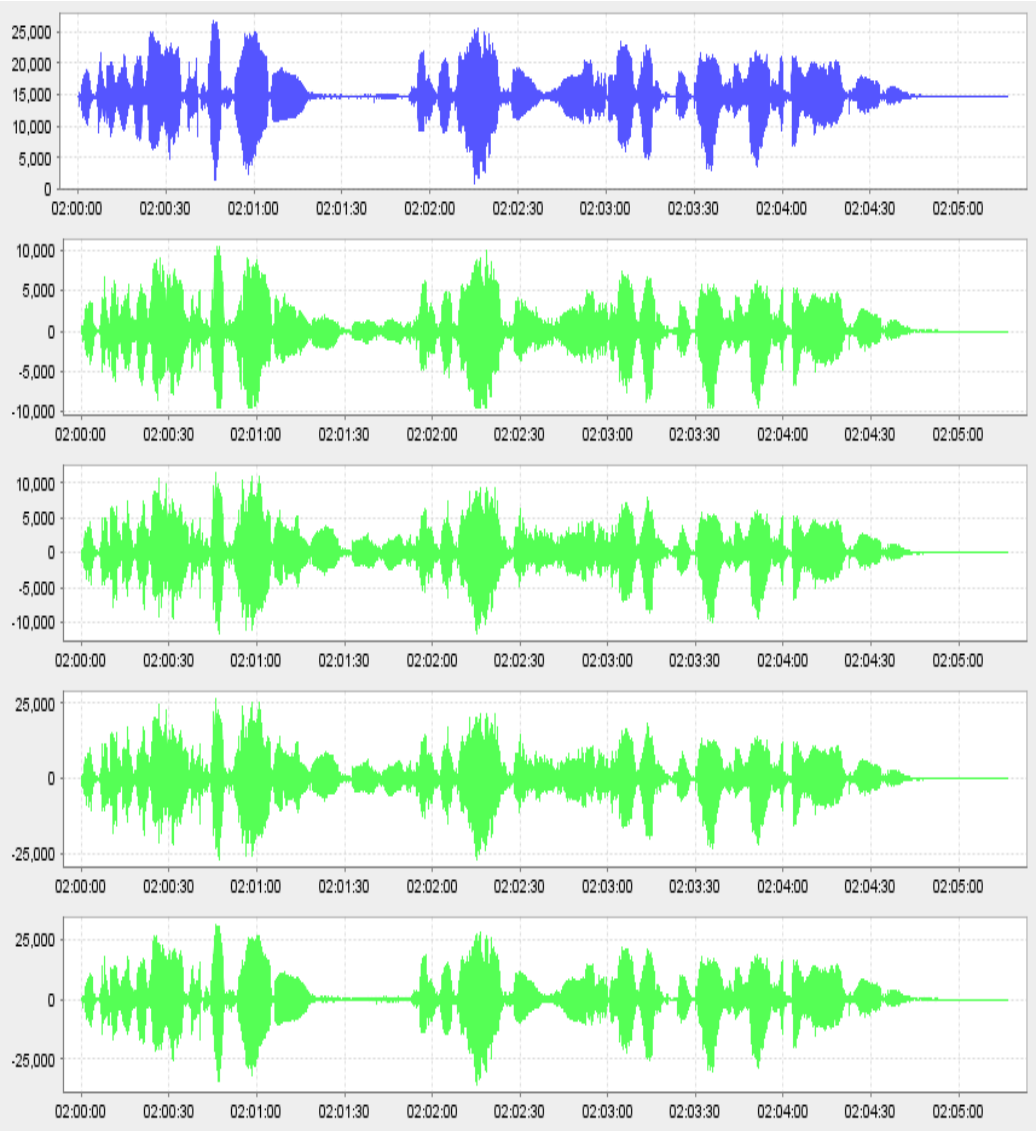

Figure 3.12: The estimations of first sources for each number of slices. Top to bottom; Original Source 1, Estimated Source 1 with 1 slice, Estimated Source 1 with 2 slices, Estimated Source 1 with 3 slices, Estimated Source 1 with 4 slices, Estimated Source 1 with 5 slices.

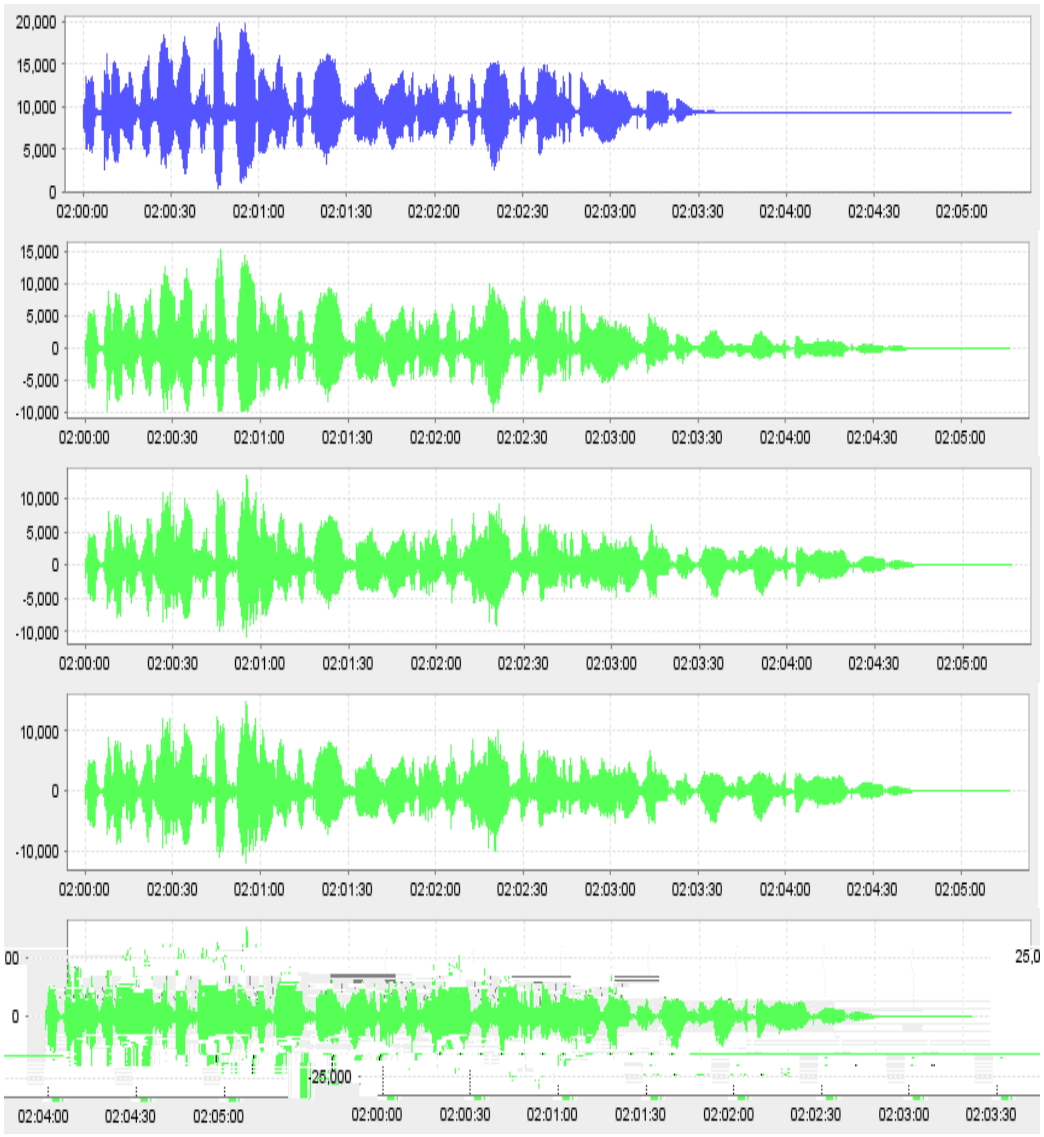

Figure 3.13: The estimations of second sources for each number of slices. Top to bottom; Original Source 2, Estimated Source 2 with 1 slice, Estimated Source 2 with 2 slices, Estimated Source 2 with 3 slices, Estimated Source 2 with 4 slices, Estimated Source 2 with 5 slices.

To investigate the effect of slice number in detail we have extended this test case by using another source set. The mixtures of these new sources are reconstructed by using ALS algorithm and beta-divergence. The reconstructed sources and the original sources are ilusturated in Fig.3.14. For both ALS algorithm and beta-divergence the 3 slice structure is constructed and the algorithms are run with indefinite loop count. The stopping condition is chosen as the convergence of Amari index. The ALS algorithm converged in 1428 iterations with an Amari index of 0.0252 and the beta divergence converged with 1021 iterations with an Amari index of 0.0142. As it is

seen in Fig.3.14 the reconstruction is evaluated succesfully for 3-slice case on the other hand the iteration number saliently increased.

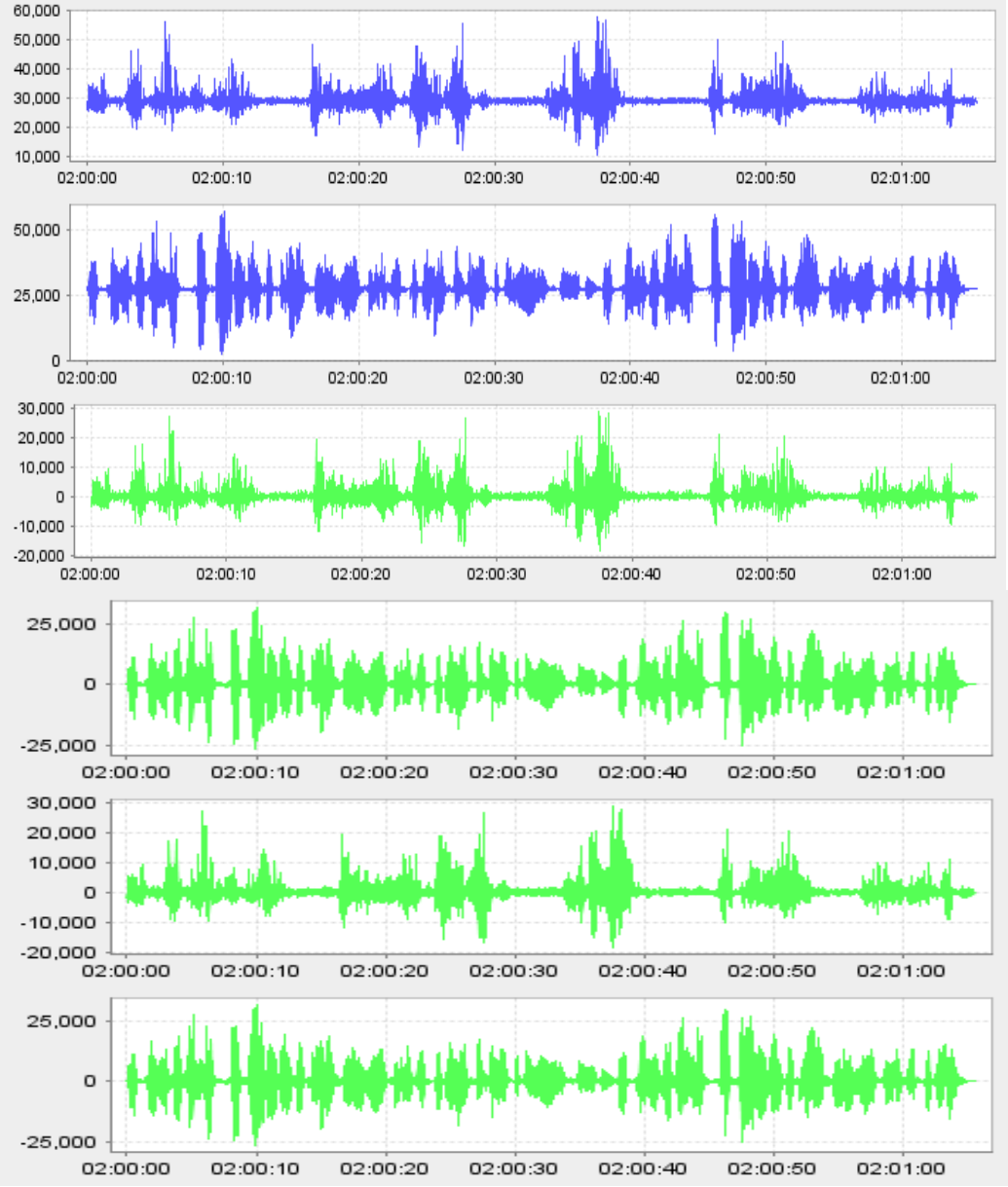

Figure 3.14: Three slice tests for beta-divergence and ALS algorithms with new source set. Top to bottom; Original Source 1, Original Source 2, Reconstructed Source 1(ALS), Reconstrurcted Source 2 (ALS), Reconstructed Source 1 (beta-div.), Reconstructed Source 2 (beta-div.)

Also for the new set of mixtures given above, the one slice case is studied as well. The ALS algorithm is run with indefinate number of iterations to see the convergence speed and as it is expected relatively accpetable recontruction is

obtained after 10880 iterations with Amari index of 0.058, Fig.3.15. The waveforms of the mixtures for 1-slice and 3-slice cases are given in Fig.A.4-A.6.

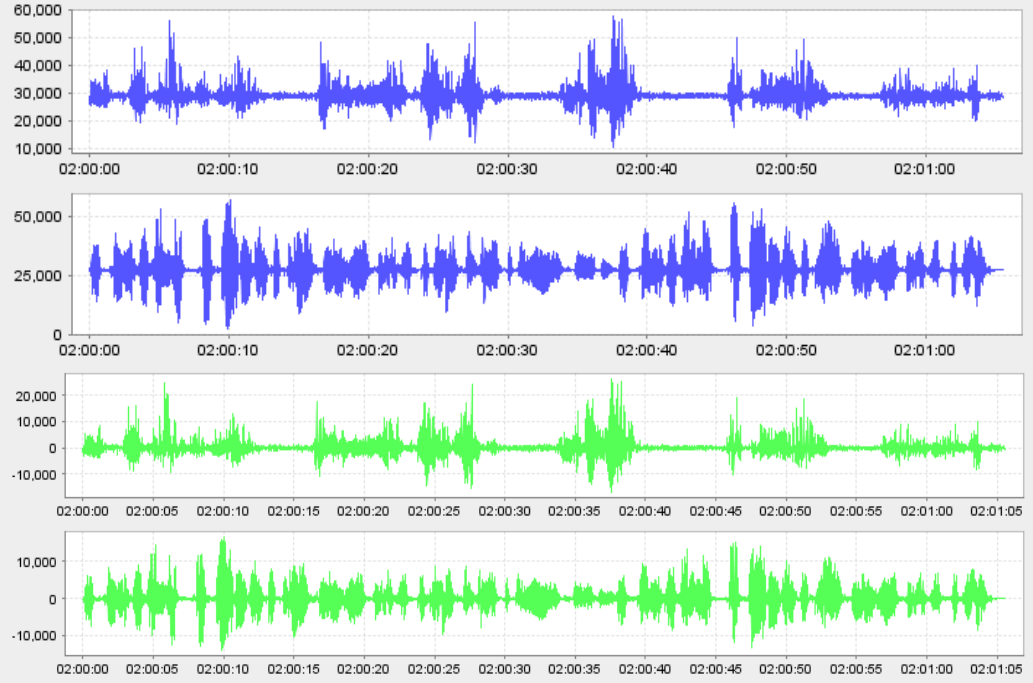

Figure 3.15 : One-Slice test for ALS with new source set. Top to bottom; Original Source 1, Original Source 2, Reconstructed Source 1, Reconstructed Source 2.

The 3-dimensional structure, obtained with K>1, is the reason of using NTF however it should be noted that the 3D-NTF2 model is used in all the test cases to be able to represent tensors by matrices. This gives the flexibility to apply the standard NMF algorithms on NTF and compare their performances.

3.2.5 Estimation Performance on Speech-Music Mixture Sets

The mixtures used in all the tests so far, are generated using 2 source signals which are speech samples of two female speakers pronouncing the same phrase. This means that the 2 source signals are assumed to be higly correlated. When this is the case, it is expected that the separation is more difficult. In this part of the work, the performance of the ALS algorithm on the mixtures which are created using one speech sample and one orchestra sample, is inspected. The three different orchestra samples are mixed with the same speech sample and three different set of mixtures are obtained. The results of the separation is given in Table 3.6. It is observed that

the Amari performance index of the first and the third mixture sets are approximately same with Amari indices obtained in the tests in which only speech-speech mixtures are used. However for the second set of mixtures, better estimation performance is acquired.

Table 3.6 : The estimation performance of the ALS algorithm, on Speech+Music mixture sets.

Mixtures	Reconstruction Error	Amari Index	Number of Iterations to Convergence
Orch1+Speech1	1.60E+07	0.03949781	38
Orch2+Speech1	82.51104965	0.01240286	32
Orch3+Speech1	0.090888319	0.03950384	39

In Fig.3.16, the change of reconstruction error and amari index through the iterations are given. It should be noted that in this test the Amari index is more distincitive than the reconstruction error, as a measure of performance.

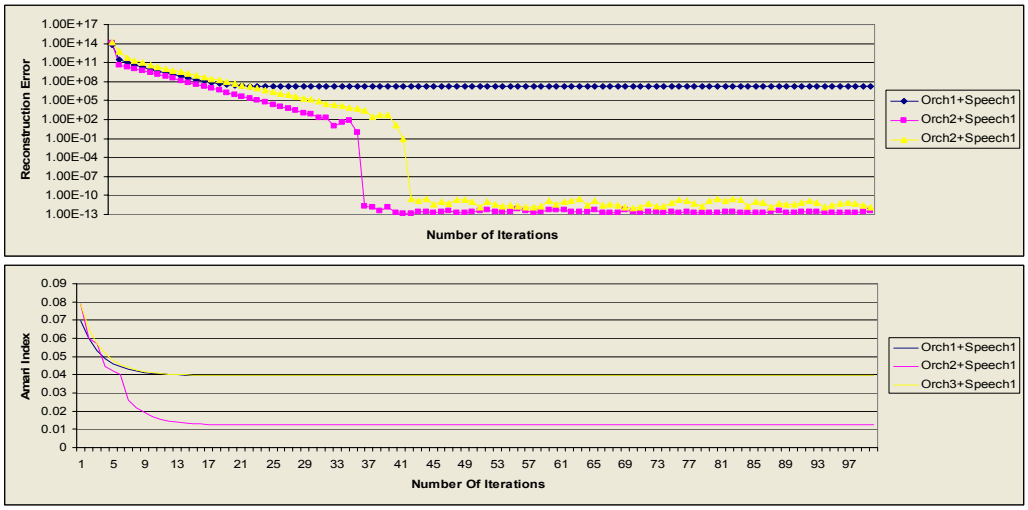

Figure 3.16 : The change of reconstruction error (top) and Amari index (bottom) through the iterations for each Speech-Music mixture set.

In Fig.3.17, the estimated waveforms of the sources for the first set of mixtures are given and also the mixture waveforms are given in Fig.A.2. It is quite hard to distinguish the difference between the orignal and the estimated sources even for the first set of mixtures whose Amari index is more poor than the second set of mixtures. The deficiency of the estimation can more clearly be seen in the second source and its estimation. Especially in the points of silence the effect of the first source over the second is obvious.

For the second set of mixtures given in Fig.A.3, the Amari index acquired is equal to 0.012. Therefore, even in the points of silence the estimated sources are almost identical to original sources, Fig.3.18. The worst Amari index is obtained for the third set of mixtures. The poor estimation of waveform of the second source can be seen in Fig.3.19. The Amari indices of the estimations for the first and the third set of mixtures are almost the same, on the other hand the estimated waveforms of the second sources are quite different. This also shows that for correct evaluation of the results all the estimated sources should be taken into account. Meaning that, even if one of the sources can be estimated perfectly this is not guaranteed for all the sources. This is the very reason of having almost the same Amari index for the third and the first set of mixtures while having different waveforms for the second sources, it is the estimations of the first sources that balance the Amari index.

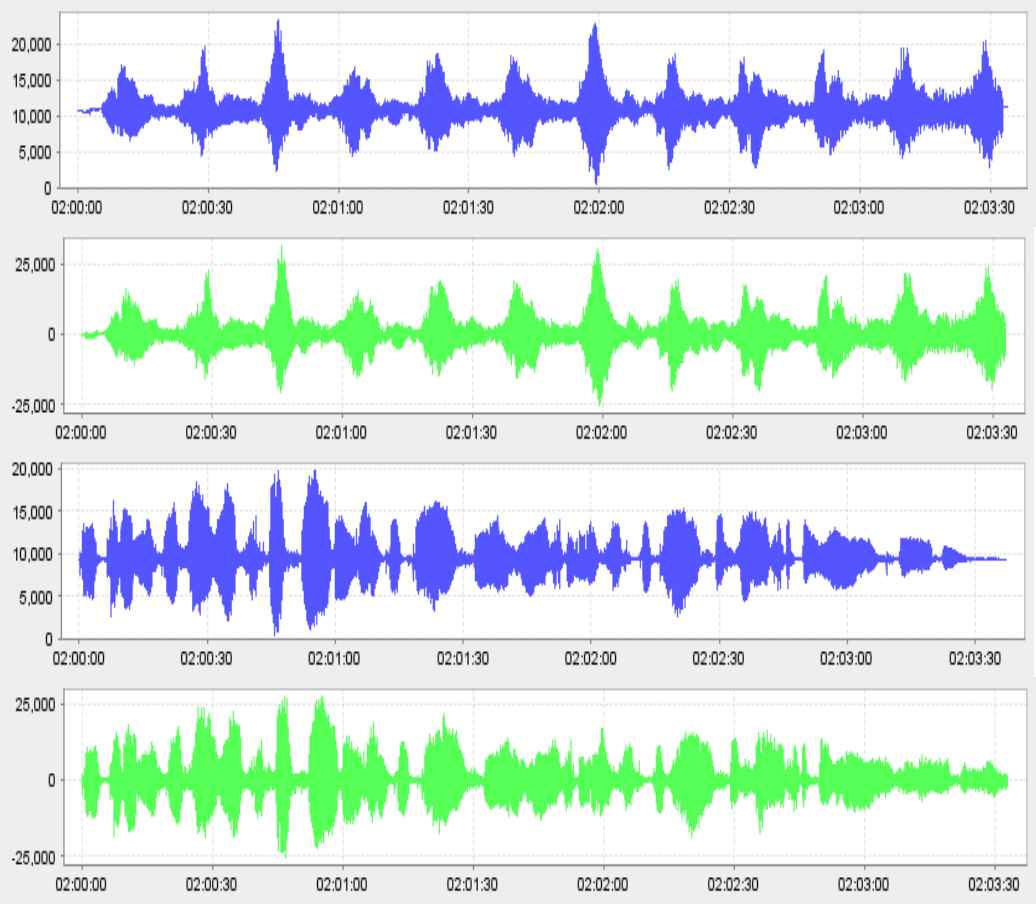

Figure 3.17 : The estimated waveforms for the 1st set of speech-orchestra mixtures. Top to bottom; Original Source 1, Estimated Source 1, Original Source 2, Estimated Source 2.

As it is mentioned before, the waveforms can sometimes be misleading. By means of waveforms, even if the estimated waveform is seemed to be quite good, it actually may not be good enough when it is listened. This issue is discussed in detail, in the subsequent sections.

Also another case which is studied here is the reconstruction of one speech source in the presence of single tone sinusoid signal in the audible frequencies. For the test of this scenario the one source is chosen as a speech signal and the other as a sinusoidal wave of 1kHz frequency. The 2-slice structure of mixtures are created using the software deisgned and the ALS algorithm is run and the Amari index of 1000 is reached after 1000 iterations.The waveforms of the reconstructed signals are given in Fig.3.20 and the mixture waveforms are given in Fig.A.5**.**

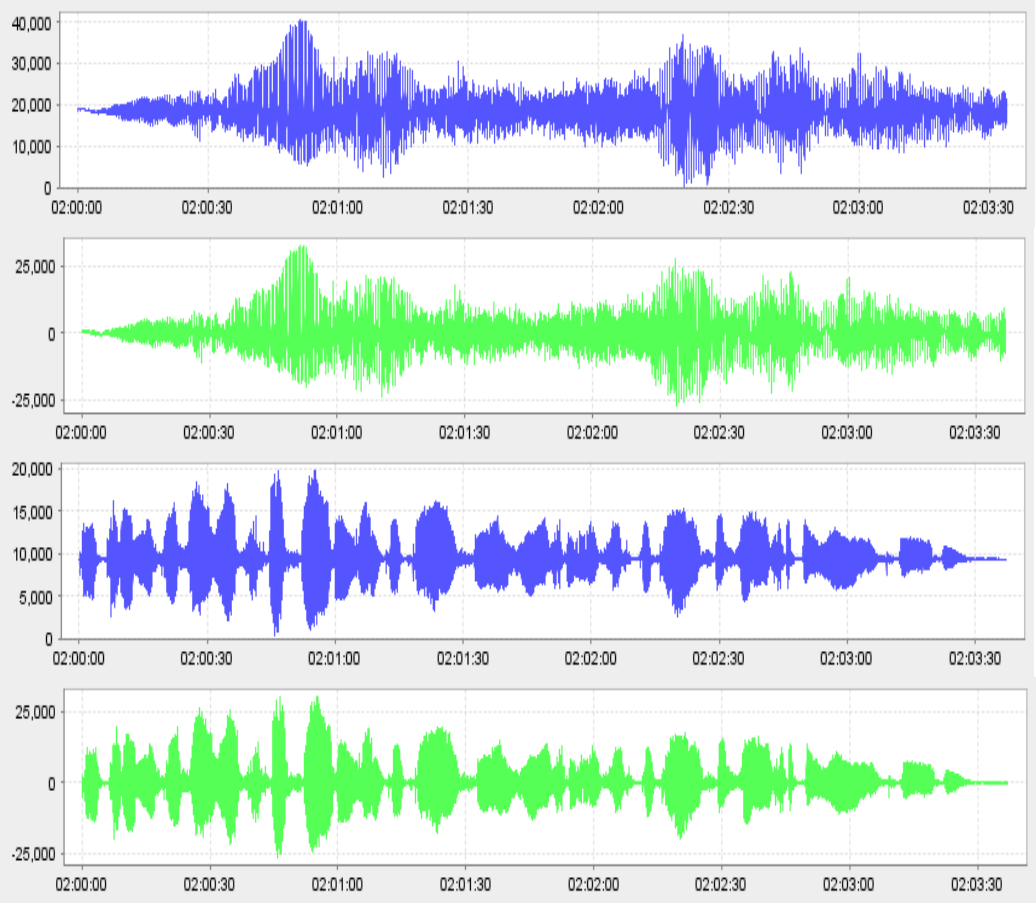

Figure 3.18 : The estimated waveforms for the 2nd set of speech-orchestra mixtures. Top to bottom; Original Source 1, Estimated Source 1, Original Source 2, Estimated Source 2.

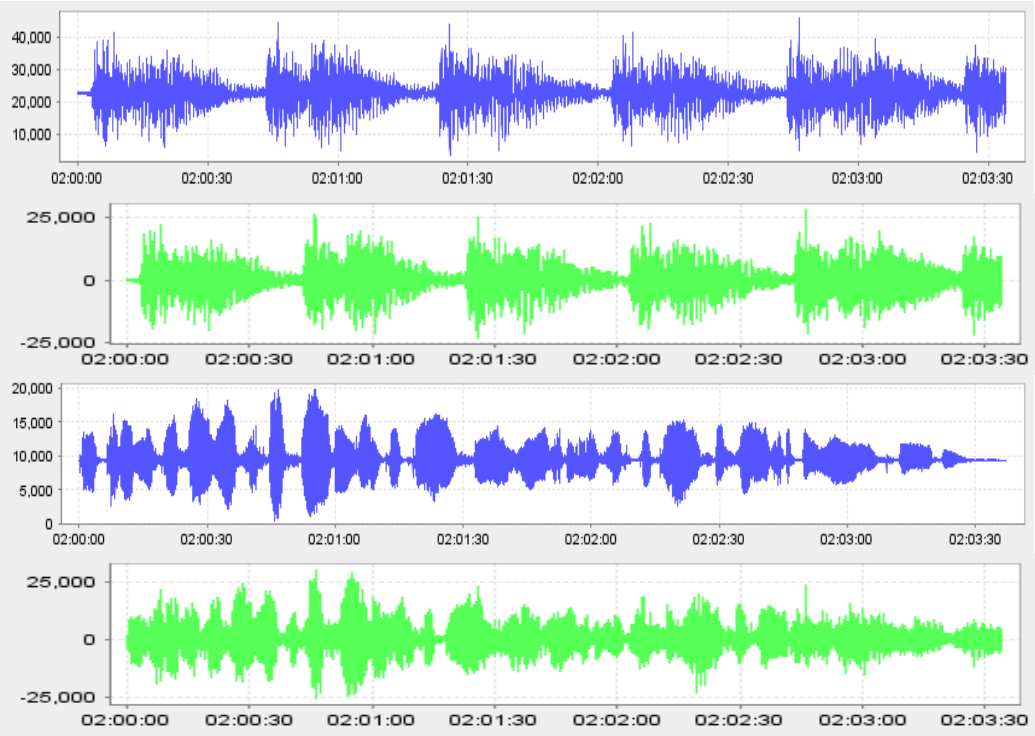

Figure 3.19 : The estimated waveforms for the 3rd set of speech-orchestra mixtures. Top to bottom; Original Source 1, Estimated Source 1, Original Source 2, Estimated Source 2.

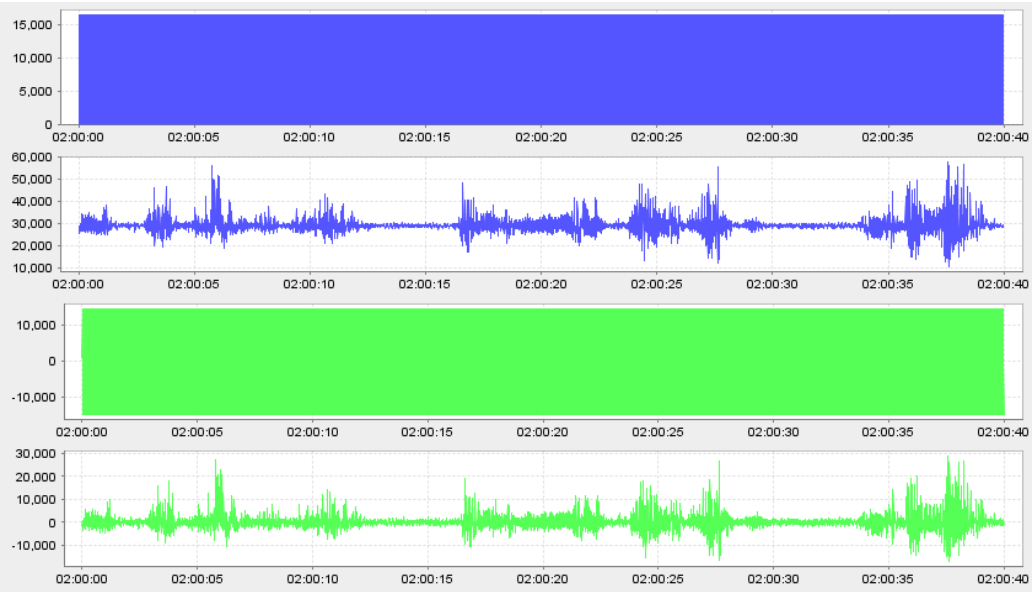

Figure 3.20 : Top to bottom; Original Source 1, Original Source 2, Reconstructed Source 1 (1kHx Sine Wave), Reconstructed Source 2.

3.2.6 Effect of Noise on the Estimation Performance

Another important problem in BSS is the noise present in the environment in which the recordings are made. Since the mixtures used in the tests are all created by linealy multiplying by a mixing matrix but not by recording, there is no noise present in our mixtures naturally. Therefore the noise is added to the mixtures after they are created. For the following tests, additive white Gaussian noise with 20SNR is used and the algorithms are run with different parameter options.

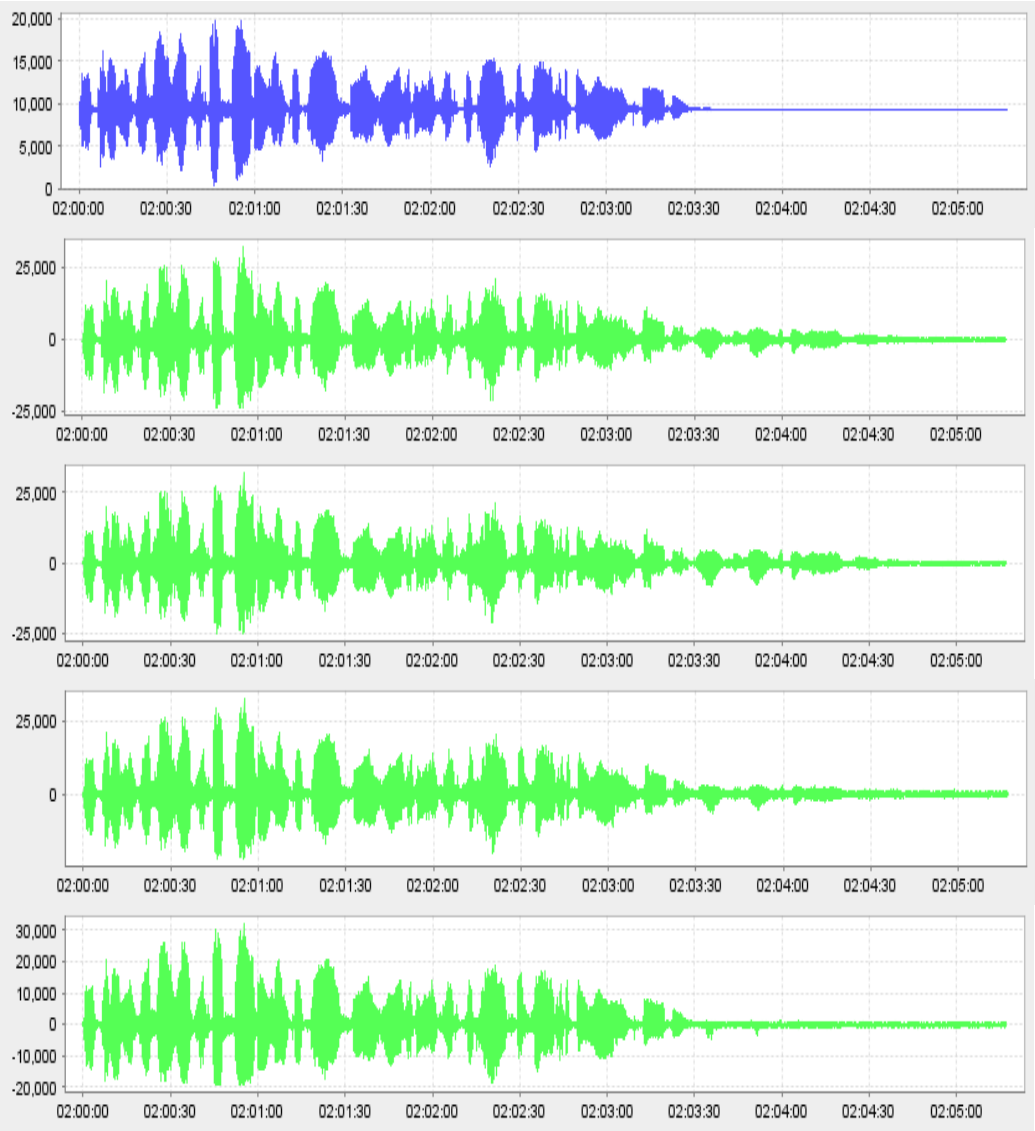

Figure 3.21 : The estimated waveforms for the noisy mixtures. Top to bottom; Original Source 1, Estimated Source 1 with RALS algorithm, Estimated Source 1 with ALS algorithm, Estimated Source 1 with Alpha algorithm, Estimated Source 1 with Beta algorithm.

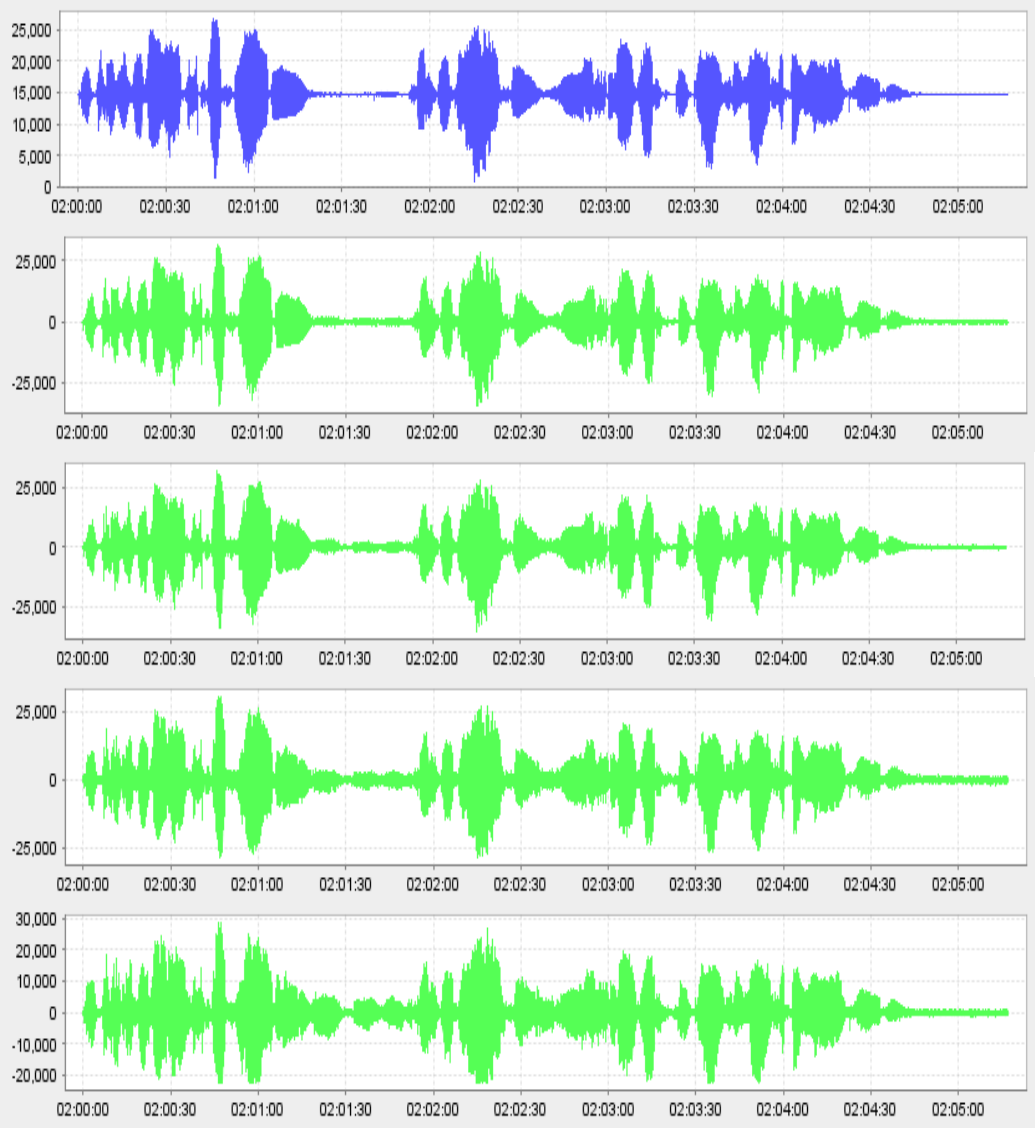

Figure 3.22 : The estimated waveforms for the noisy mixtures. Top to bottom; Original Source 2, Estimated Source 2 with RALS algorithm, Estimated Source 2 with ALS algorithm, Estimated Source 2 with Alpha algorithm, Estimated Source 2 with Beta algorithm.

The best performance is acquired with the beta algorithm, however as it is seen in Table 3.7 the Amari index of the best estimation is even quite poor. Here, the waveforms of the estimated sources plays important role to evaluate the results. The Amari index of the best estimations for each algorithm is given in Table 3.7.

Table 3.7 : The estimation performance of the algorithms, on noisy mixtures.

	RALS	ALS	Alpha	Beta
Best Amari Index	0.162	0.196	0.099	0.088

By only looking at the Amri indices, it can be thought that the separation is not performed successfully, whereas the waveforms of estimated sources shown in Fig.3.21 and Fig.3.22 reveal the fact that estimated sources are similar to original sources. The reason of having poor Amri indices for these estimations is assumed to be the noise which is still present on the estimated sources. Therefore it can be deduced that algorothms can still be run for noisy mixtures however there is no noise reduction is acquired on the estimated sources.

3.2.7 Objective Evaluation of Perceived Audio Quality

The traditional measures which are used to evaluate the decomposition performance are known to be not informative about perceptual quality of the audio signals. Therefore performance of the designed system is evaluated not only with the well known Amari index, but also with perceptual audio quality criterions which are defined in the recommendation report, ITU-R BS.1387 [27] of International Telecommunication Union (ITU).

The perceived quality of audio signals are used to be evaluated by means of subjective listening tests. In the recommendation report of ITU, the objective evaluation process of audio quality is stated with set of standards in all aspects. In BS.1387, there are two ear models, one of which is based upon filter banks and the other on Fast Fourier Transform (FFT), are proposed. Also there are 13 features, defined to model several physiologic and psycho-acoustic effects. These features are also called Model Output Variables (MOV) and some of them are calculated by using FFT model whereas the others by using filter bank model, as shown in Table 3.8. The short description and mathematical representation of MOVs can be given as follows. More detailed explanations are given in [27,28].

BandwidthRef and BandwidthTest: The bandwidth of the reference and test signals, respectively. It is calculated by averaging the instantaneous bandwidth of each frame as given in Eq.3.4. [27,28]

$$W_{T,R} = \frac{1}{N}\sum_{n=0}^{N-1} K_{T,R}[n], \qquad \textbf{(3.4)}$$

where N is the total number of frames and $K_{T,R}[n]$ is the bandwidth of the n-th frame.

Table 3.8 : The Model Output Variables

MOV	Definition	Model
BandwidthRef	The bandwidth of the reference signal	FFT
BandwidthTest	The bandwidth of the test signal	FFT
Total NMR	Noise to mask ratio	FFT
WinModDiff	Windowed modulation difference	FFT
ADB	Average block distortion	FFT
EHS	Harmonic structure of the error	FFT
AvgModDiff1	Average modulation difference	FFT
AvgModDiff2	Average modulation difference	FFT
RmsNoiseLoud	Distortion loudness	FFT
MFPD	Maximum filtered probability of detection	FFT
Segmental NMR	Segmental Noise to mask ratio	FFT
RmsModDiff	Modulation changes	Filter Bank
RmsNoiseLoudAsym	Distortion loudness	Filter Bank

Noise to Mask Ratio (NMR): The ratio of noise to the masking threshold. Here, noise represents the difference between the amplitudes of the test and reference signals. The total NMR is calculated as in Eq.3.5. [27,28]

$$R_{NMtot} = 10\log_{10}\left(\frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{N_c}\sum_{k=0}^{N_c-1}R_{NM}[k,n]\right), \quad \textbf{(3.5)}$$

where N is the total number of frames, N_C is the total number of critical bands and $R_{NM}[k,n]$ is the noise to mask ratio in the n-th frame and k-th band.

WinModDiff: This feature is calculated by averaging the modulation difference between the test and reference signals for each frame as given in Eq.3.6. [27,28]

$$M_{WDiff1} = \sqrt{\frac{1}{N-L+1}\sum_{n=L-1}^{N-1}\left(\frac{1}{L}\sum_{i=0}^{L-1}\sqrt{\widetilde{M}_{diff1}[n-i]}\right)^4}, \quad \textbf{(3.6)}$$

where L is the number of sliding frames and $\widetilde{M}_{diff1}[n]$ is the scaled modulation difference of each frequency band.

AvgModDiff1: This feature is defined to measure the average modulation difference between the test and the reference signal. The difference between this Mov and the

previous one arise from the weighting operation used while averaging. The corresponding mathematical expression is given in Eq.3.7. [27,28]

$$A_{Diff1} = \frac{\sum_{n=0}^{N-1} W_1[n]\widetilde{M}_{diff1}[n]}{\sum_{n=0}^{N-1} W_1[n]}, \quad \textbf{(3.7)}$$

where $W_1[n]$is the temporal weighting term which is calculated by using the modulation pattern loudness of the reference signal.

AvgModDiff2: As it is given in Eq.3.8. This MOV is the same as the previous one, except the weighting term, $W_2[n]$ which is calculated by using the internal noise term**.** [27,28]

$$A_{Diff2} = \frac{\sum_{n=0}^{N-1} W_2[n]\widetilde{M}_{diff2}[n]}{\sum_{n=0}^{N-1} W_2[n]}. \quad \textbf{(3.8)}$$

RmsModDiff: This feature is defined as the square of the scaled modulation difference for each frequency band and given in Eq.3.9. [27,28]

$$M_{Diff} = \sqrt{N_c \frac{\sum_{n=0}^{N-1} \left(W_A[n]\widetilde{M}_{Diff}[n]\right)^2}{\sum_{n=0}^{N-1} \left(W_A[n]\right)^2}}. \quad \textbf{(3.9)}$$

MFPD: The measure of the probability of detecting the differences between reference and test signals. [27,28]

$$MFPD = \widetilde{P}_M[N-1], \quad \textbf{(3.10a)}$$

$$P_M[n] = \begin{cases} P_b[n], & P_b[n] > P_M[n-1] \\ P_M[n-1], & P_M[n-1] > P_b[n] \end{cases}, \quad \textbf{(3.10b)}$$

where N is the total number of frames, $P_M[n]$ is the maximum filtered detection probability of the n-th frame and $P_b[n]$ is the total probability of detection.

ADB: The average distorted block MOV measures the total number of steps above the threshold, Q_S Eq.3.11. [27,28]

$$\mathrm{ADB} = \begin{cases} 0 & \mathrm{N} = 0, \\ \log_{10}(\mathrm{Q_s}/\mathrm{N}) & \mathrm{N} > 0 \wedge \mathrm{Q_S} > 0, \\ -0.5 & \mathrm{N} > 0 \wedge \mathrm{Q_S} = 0. \end{cases} \quad \textbf{(3.11)}$$

EHS: This feature states the harmonic structure of error as given in Eq.3.12 and it is calculation is similar to cepstrum analysis. [27,28]

$$\mathrm{E_H} = \frac{1000}{\mathrm{N}} \sum_{\mathrm{n}=0}^{\mathrm{N}-1} \mathrm{E_{H\,max}}(\mathrm{n}), \quad \textbf{(3.12)}$$

where N is the total number of frequency bins in the power spectrum and $E_{H\,max}(n)$ is the peak value of the correlation power spectrum.

Segmental NMR: This feature is calculated by averaging the Noise to Mask ratios of each frame as given in Eq.3.13. [27,28]

$$\mathrm{R_{NMSeg}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{n}=0}^{\mathrm{N}-1} 10 \log_{10} \left(\frac{1}{\mathrm{N_C}} \sum_{\mathrm{k}=0}^{\mathrm{N_C}-1} \mathrm{R_{NM}}[\mathrm{k},\mathrm{n}] \right), \quad \textbf{(3.13)}$$

where N is the total number of frames and N_C is the total numbers of critical bands.

RmsNoiseLoud: Defined as the squared average of the instantaneous noise loudness, given in Eq.3.14. [27,28]

$$\mathrm{N_{Lrms}} = \sqrt{\frac{1}{\mathrm{N}} \sum_{\mathrm{n}=0}^{\mathrm{N}-1} \left(\widetilde{\mathrm{N}}_{\mathrm{L}}[\mathrm{n}] \right)^2}, \quad \textbf{(3.14)}$$

where $\widetilde{N}_L[n]$ is the instantaneous noise loudness of n-th frame.

RmsNoiseLoudAsym: This is the sum of the squared average of instantaneous noise loudness(N_{Lrms}) and the loudness of missing components in test signals(N_{Mrms}), given in Eq.3.15. [27,28]

$$N_{LM} = N_{Lrms} + 0.5N_{Mrms} . \quad \textbf{(3.15)}$$

Once these features are extracted, a feature vector is composed and classified into one of five different perceptual quality class [27,28]. These classes states the objective perceived quality difference between the reference signal and the test signal as; Imperceptible / Perceptible, not annoying / Slightly annoying / Annoying / Very annoying. In BSS situation, the reference signals are the source signals and the test signals are the signals that are obtained after separation.

The evaluation of perceptual quality of the separated signals are given in this part of the thesis. The tests are performed for two cases as given in Table 3.9 and Table 3.10. These tests are; the effect of using different set of mixtures and the effect of different number of mixing matrices, on the perceived audio quality. The separation is performed by optimizing the β-divergence cost function and β and α values are set experimentally as 1 and 100, respectively. The objective perceived audio quality measurements are performed by using the software called OPERA Software Suite V3.0 which is developed by OPTICOM.

Table 3.9 : Perceptual Quality versus Amari index for different set of mixtures.

Observation Set	Amari Index	Perceptual Quality (Source 1)	Perceptual Quality (Source 2)
1	0.0193707	Perceptible, not annoying	Slightly annoying
2	0.0157731	Perceptible, not annoying	Perceptible, not annoying
3	0.0111056	Imperceptible	Slightly annoying
4	0.0202030	Annoying	Slightly annoying
5	0.0336697	Annoying	Annoying

Table 3.10 : Perceptual Quality versus Amari index for different number of mixing matrices.

Number of Mixing matrices(Slices)	Amari Index	Perceptual Quality (Source 1)	Perceptual Quality (Source 2)
1(NMF)	0.1275634	Annoying	Annoying
4	0.0177651	Perceptible, not annoying	Slightly annoying

It has been observed that some of the decomposed sources are acceptable according to Amari index while they are not with respect to the perceptual quality criteria thus it can be concluded that the perceptual criteria is more suitable to objective quality evaluation of audio.

4. CONCLUSION AND FUTURE WORK

The purpose of this study is to research the performance of NTF techniques in blind audio source separation which is also refered to as ‘cocktail party problem’. The performance of three differenet NTF algorthms, proposed in [10-13], are investigated under various test conditions which includes mixing, initialization, noise, regularization, sparseness cases. It has been observed that in general the performace of the NTF algorithms are quite promising in blind audio source separation and even more successful than some other famous methods like NMF, under certain conditions. Among these three different algorithms that are studied, it is seen that the performance of ALS algorithm, with and without the regularization, is superior than the other two. It should be noted that the performance of the beta-divergence algorithm can be competitive with the ALS algorithm if the parameters can be chosen carefully. However optimum parameter selection is another issue that must be studied. Despite the performance improvement that can be obtained in beta-divergence algorithm with optimum parameter selection, the computational complexity is still present as a disadvantage.

In the noisy mixture case, it is observed that the separation is performed successfully however the noise is still present in the estimated signals. Therefore we can deduce that the separation algorithms are robust to noise but noise reduction is not fulfilled in the estimated signals. The initialization of the algorithms are performed randomly throughout the tests and it is seen that even with random initialization the perfomance of the separation is high. The research of algorithm performances which are initialized using different approaches, are left as a future work. It is important to note that, all these interpretations are made upon the Amari index results obtained in the tests, whereas it is given in perceived audio quality tests that only considering Amari index does not give adequate information about the separation performance. Therefore it is suggested that perceived audio quality criterions which are proposed by ITU in ITU-REC BS.1387, should also be taken into account. Consequently, it has been experienced that the NTF methods are successful in blind audio source

separation and it is an open field that deserves in depth research. The representation of audio data and optimum parameter selection for regularization and and sparseness are decided to be the important issues that must be studied in future.

REFERENCES

[1] **Manuel Reyes-Gomez, Nebojsa Jojic, Daniel P.W. Ellis,** 2004, Towards single-channel unsupervised source separation of speech mixtures: The layered harmonics / formants separation - tracking model, *SAPA*.

[2] **Pia Anttila, Pentti Paatero**, **Unto Tapper, Olli Järvinen,** 2002, Estimation of the mixing matrix for underdetermined blind source separation using spectral estimation techniques, *in Proceedings Eusipco vol. I*, 557–560.

[3] **Jollife**, 1986. Principle Component Analysis, *Springer-Verlang*, New York.

[4] **Comon P.**, 1994. Independent Component Analysis, a new concept, Signal Processing, *Elseiver* **36**(3): 287-317.

[5] **Golub, Gene H., Kahan, William**, 1965. Calculating the singular values and pseudo-inverse of a matrix, *Journal of the society for industrial and applied Mathematics* : Series B, Numerical Analysis 2: 205-224.

[6] **Paatero P., Tapper U.**, 1994. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, *Environmetrics* **5**: 111-126.

[7] **Pia Anttila, Pentti Paatero**, **Unto Tapper, Olli Järvinen,** 1995. Source identification of bulk wet deposition in Finland by positive matrix factorization, *Atmospheric Environment* **29** (14): 1705–1718.

[8] **Daniel D. Lee and H. Sebastian Seung**, 1999. Learning the parts of objects by non-negative matrix factorization, *Nature* **401** (6755): 788–791.

[9] **Daniel D. Lee and H. Sebastian Seung**, 2001. Algorithms for Non-negative Matrix Factorization, *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, 556-562, MIT Press.

[10] **De Lathauwer L. and Comon P. (Eds.),** 2005. Workshop on Tensor Decompositions and Applications, *CIRM*, Marseille, France.

[11] **Heiler M. and Schnoerr C.**, 2006. Controlling sparseness in nonnegative tensor factorization, in *ECCV*, Springer LNCS, vol.3951, 56–67.

[12] **Smilde A., Bro R., and Geladi P.**, 2004. Multi-way Analysis: Applications in the Chemical Sciences, John Wiley and Sons, NewYork.

[13] **Berry M., Browne M., Langville A., Pauca P., and Plemmons R.**, 2007 Algorithms and applications for approximate nonnegative matrix factorization, Computational Statistics and Data Analysis.

[14] **Cichocki A., Zdunek R., and Amari S.**, 2006. Csiszar's divergences for non-negative matrix factorization: Family of new algorithms, Springer LNCS, vol.3889, 32–39.

[15] **Cichocki A., Amari S., Zdunek R., Kompass R., Hori G., and He Z.**, 2006. Extended SMART algorithms for non-negative matrix factorization, Springer LNAI, vol. 4029, 548–562.

[16] **Cichocki A. and Zdunek R.**, 2006. NTFLAB for Signal Processing, *Tech. Rep., Laboratory for Advanced Brain Signal Processing*, BSI, RIKEN, Saitama, Japan.

[17] **Dhillon I. and Sra S.**, 2005. Generalized nonnegative matrix approximations with Bregman divergences, *in Neural Information Proc. Systems*, Vancouver, Canada.

[18] **Morup M., Hansen L. K., Herrmann C. S., Parnas J., and Arnfred S. M.**, 2006. Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG, *NeuroImage*, vol.29, no.3, 938–947.

[19] **Miwakeichi F., Martnez-Montes E., Nishiyama P. A. N., Mizuhara H., and Yamaguchi Y.**, 2004. Decomposing EEG data into spacetime-frequency components using Parallel Factor Analysis, *NeuroImage*, vol. 22, no.3, 1035–1045.

[20] **Amari S.**, 1985. Differential-Geometrical Methods in Statistics, Springer Verlag,.

[21] **Hancewicz, T. M., Wang, J. H.**, 2005. Discriminant image resolution: a novel multivari ate image analysis method utilizing a spatial classification constraint in addition to bilinear nonnegativity. *Chemometrics and Intelligent Laboratory Systems* 77 18–31

[22] **Minami M. and Eguchi S.**, 2002. Robust blind source separation by beta-divergence, *Neural Computation*, vol. 14, 1859–1886.

[23] **Kompass R.**, 2005. A generalized divergence measure for nonnegative matrix factorization. *Neuroinfomatics Workshop*, Torun, Poland.

[24] **Csisz´ar I.**, 1974. Information measures: A critical survey, *Conference on Information Theory*, Academia Prague, volume A, 73–86.

[25] **German Gomez-Herrero, Atanas Gotchev, Karen Egiazarian**, 2005. Distortion Measures for Sparse Signals, *International Conference on Computer Systems and Technologies - CompSysTech*

[26] **Fabian J. Theis , Gonzalo A. Garcia**, 2005. On the use of sparse signal decomposition in the analysis of multi-channel surface electromyograms, Elsevier Science.

[27] **Kabal P.**, 2002. An examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality, *Telecommunications and Signal Processing Laboratory Technical Report*, McGill University.

[28] **ITU-R Recommendation BS.1387**, 2001. Methods for Objective Measurements of Perceived Audio Quality, *International Telecommunication Union*.

APPENDIX A

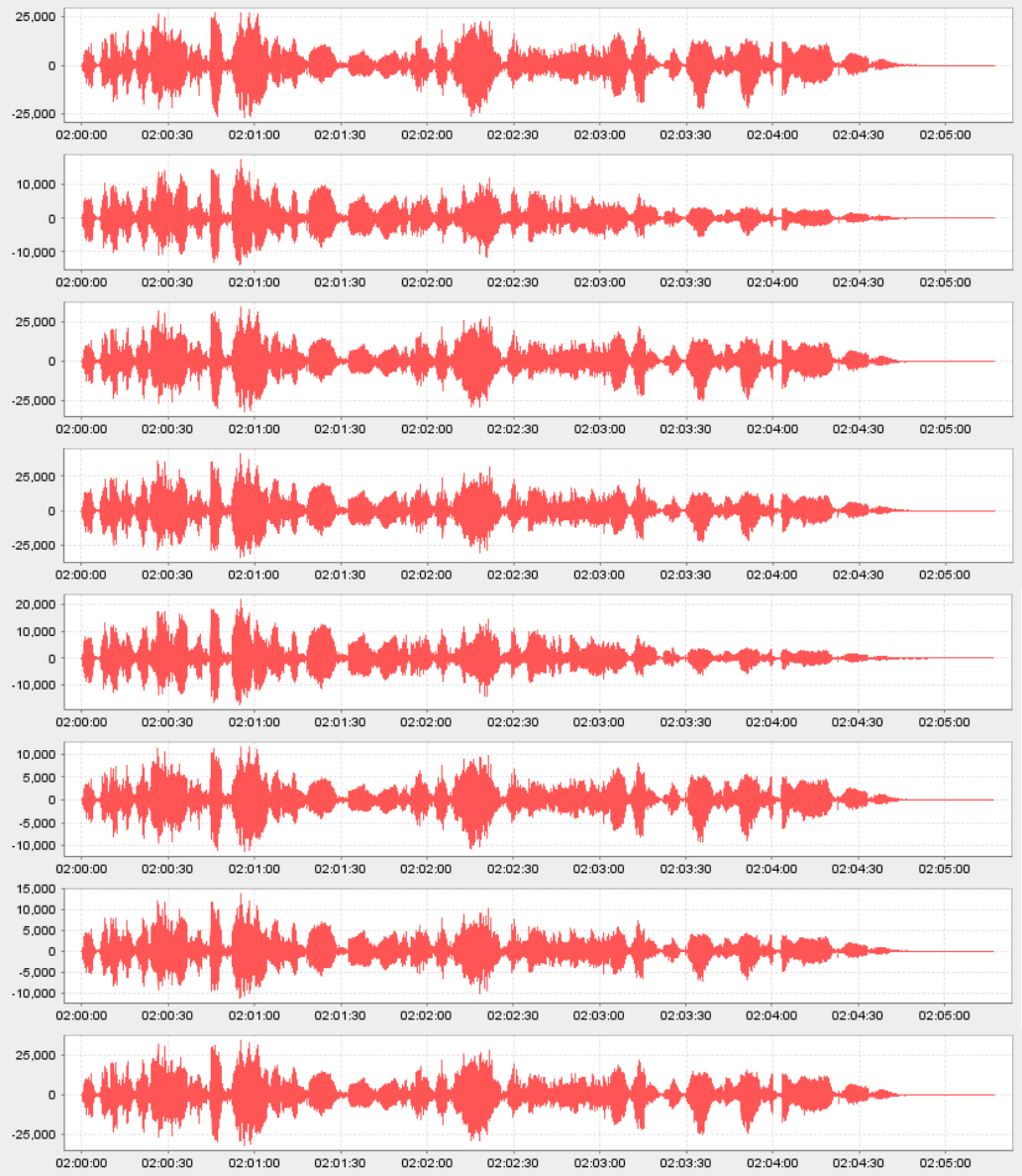

Figure A.1 : Mixtures of Speech Source 1 and Speech Source 2

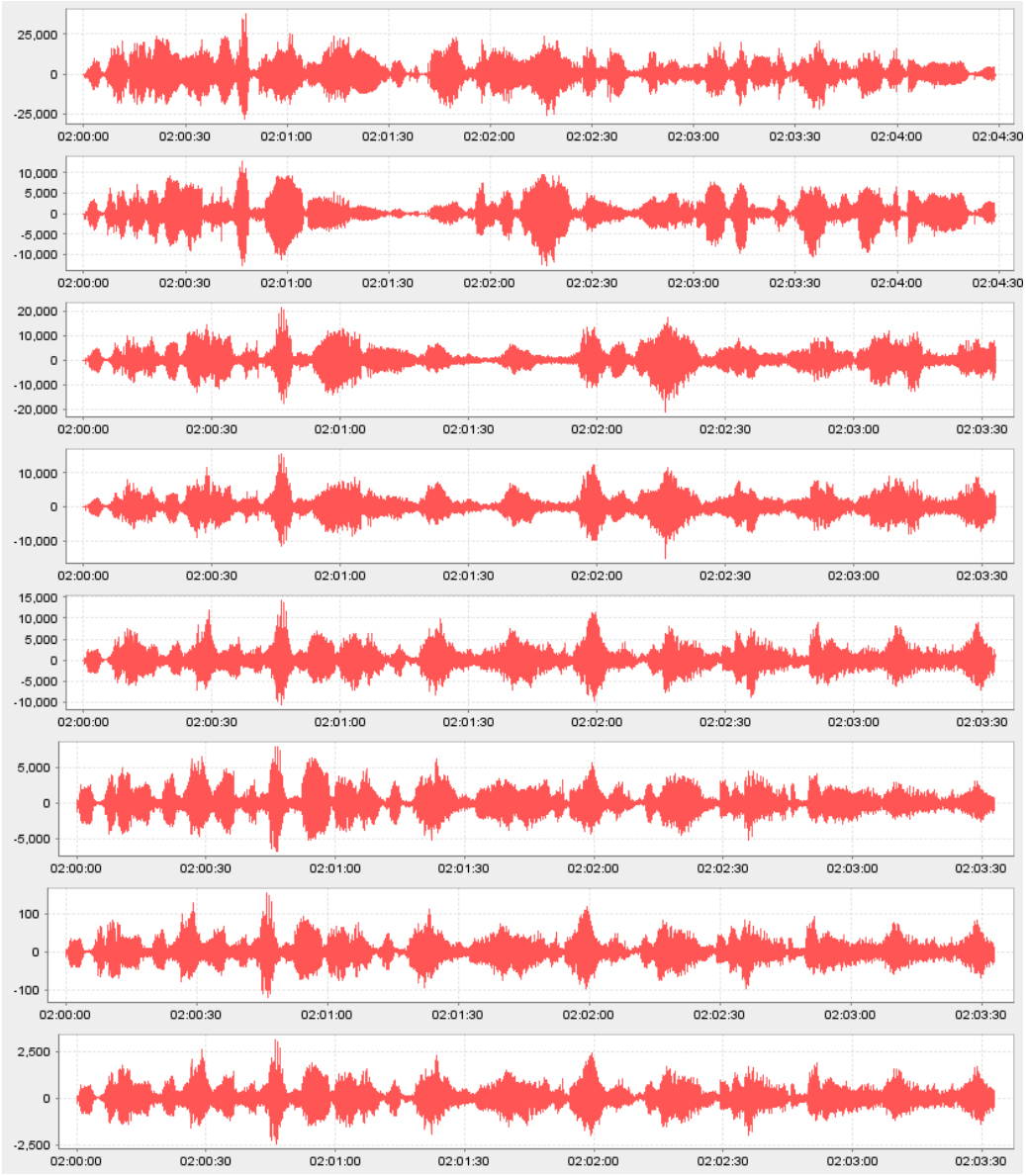

Figure A.2 : Mixtures of Speech Source 1 and Orchestra Source 1

Figure A.3 : Mixtures of Speech Source 1 and Orchestra Source 2

Figure A.4 : Mixtures of Speech Source 3 and Speech Source 2 for extended 1-Slice test

Figure A.5 : Mixtures of Speech Sinusiodal Signal and Speech Source 3

Figure A.6 : Mixtures of Speech Source 3 and Speech Source 2 for extended 3-Slice tests

Table A.1 : 1st Mixing matrix set

0.709979	0.417498
0.730612	0.957471
0.482024	0.207019
0.512758	0.85239
0.627253	0.623825
0.830932	0.83897
0.963756	0.491891
0.947326	0.044727

Table A.2 : 2nd Mixing matrix set

0.303845	0.462612
0.243811	0.528068
0.76732	0.922727
0.033377	0.003483
0.842624	0.495601
0.283129	0.354004
0.477568	0.453138
0.031526	0.497983

Table A.3 : 3rd Mixing matrix set

0.625721	0.327052
0.628054	0.915028
0.574015	0.142777
0.404948	0.113091
0.236487	0.81792
0.057592	0.760984
0.128086	0.725873
0.602816	0.290203

Table A.4 : 4th Mixing Matrix set

0.613023	0.228541
0.44776	0.390732
0.229298	0.352845
0.335693	0.268434
0.341699	0.384139
0.498593	0.767873
0.027095	0.638998
0.628396	0.660506

Table A.5 : 5th Mixing Matrix set

0.996242	0.860237
0.952809	0.472512
0.774194	0.309832
0.425778	0.894654
0.206229	0.62176
0.297537	0.339919
0.440953	0.041295
0.371865	0.197485

CIRRICULUM VITAE

M. Altug KEYDER

Permanent Address: Atatürk cad. Ata apt. 63/39 Erenköy/İstanbul, TURKEY.
Phone: 0216 411 20 72 – Mobile: 0533 413 96 92
Email: altug13@yahoo.com
Web: http://www.mspr.itu.edu.tr/altug/altugIndex.htm

PERSONAL DETAILS

Date of Birth: 15.02.1982
Place of Birth: Istanbul, TURKEY
Nationality: Turkish
Marital Status: Single
Driver Licence: Class-B

EDUCATION

High School: Marmara Private High School
B.Sc.: Yeditepe University, Double Major (2000-2005)
Electrical and Electronics Engineering. (full scholarship),
Computer Science Engineering. (full scholarship).
M.Sc.: Istanbul Technical University (2005-2008)
Telecommunications Engineering.

PROJECTS AND RESEARCHES

Control Applications With Microcontrollers Using Speech Recognition, B.Sc. Thesis - 2005.
Blind audio Source Separation Using Non-Negative Tensor Factorization Techniques, M.Sc. Thesis - 2007.
Broadcast Monitoring and Audio Identification by Audio Fingerprinting, Research Project @ MSPR Lab. Istanbul Technical Uni. (supported by TUBITAK)

SUMMARY OF QUALIFICATIONS

Programming skills: advanced level C/C++, JAVA, Assembly(Intel MCS-51 microprocessor family), intermediate level web programming.
Platforms: Linux, MS-Windows.
Other Application Programs: MS-Office, Matlab, several Audio Signal Processing and Circuit Simulation Programs.
Languages known: English(advanced), German(beginner).

EMPLOYMENT

İris Telecommunication, intern, 2003.
TÜBİTAK - UME Time and Frequency Lab., intern, 2004.
Yeditepe University Electrical and Electronics Engineering Dept., Student Assist., 2004-2005.
Istanbul Technical University, Multimedia Signal Processing and Pattern Recognition Lab., 2005-2007.